Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AI-Assisted Regulatory Intelligence and Adaptive Compliance Verification System for Banking Supervision

Sofiya Rukadikar
COEP Technological University
Pranav Bhoite
COEP Technological University
Shripad Kulkarni
COEP Technological University
Pratiksha Deshmukh
COEP Technological University
Nisarg Wath
nisargnw24.comp@coeptech.ac.in
COEP Technological University

Research Article

Keywords:

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11450785/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

AI-Assisted Regulatory Intelligence and Adaptive Compliance Verification System for Banking Supervision

Sofiya Rukadikar
Department of Computer Engineering
COEP Technological University
Pune, India
rukadikarss24.comp@coeptech.ac.in

Shripad Kulkarni
Department of Computer Engineering
COEP Technological University
Pune, India
kulkarnisn24.comp@coeptech.ac.in

Nisarg Wath
Department of Computer Engineering
COEP Technological University
Pune, India
nisargnw24.comp@coeptech.ac.in

Pranav Bhoite
Department of Computer Engineering
COEP Technological University
Pune, India
bhoitepm24.comp@coeptech.ac.in

Pratiksha Deshmukh
Department of Computer Engineering
COEP Technological University
Pune, India
deshmukhpr@coeptech.ac.in

***Abstract*—Regulatory requirements issued by the Reserve Bank of India (RBI) fall into categories including numerical thresholds, deterministic conditions, temporal obligations, and qualitative standards such as "adequate due diligence," each of which requires a different mode of verification. Bank employees and officers must first locate and interpret the appropriate requirement among an ever-growing corpus of Master Directions, circulars, and notifications, while the regulator must independently verify (at scale) whether submitted bank evidence satisfies the requirement. We present a two-module regulatory intelligence system that addresses both sides of this problem in a single pipeline. The Regulation Comprehension module ingests an RBI circular or Master Direction and segments and categorizes its clauses, extracts a typed and machine-readable rule representation, simplifies regulatory language for reading by humans, and enables natural-language question answering with clause-level sourcing. The main technical contribution is an adaptive reasoning-modality router that determines, for a given regulatory requirement and evidence set, whether a deterministic rule engine, a retrieval-based large language model (LLM), or human escalation should be used, governed by an explicit cost–accuracy–latency objective and a hard architectural override that directs fully specified deterministic rules and complete evidence to the rule engine unconditionally. This differs from the well-studied task of routing between LLMs of differing capability, since our router chooses among fundamentally different reasoning mechanisms rather than among models. We also contribute a rule-grounded synthetic RBI compliance benchmark across nine difficulty categories and a dedicated regulation-comprehension benchmark design (with extraction, question-answering, and simplification variants), such that the two modules can be evaluated with equal rigor rather than treating comprehension as a subordinate preprocessing step, yielding two distinct sets of metrics, baselines, and ablation points, each for one of the modules, across extraction accuracy, retrieval and answer-faithfulness, simplification quality, compliance accuracy, and routing efficiency. We also contribute an evidence-grounded explanation architecture that traces every such decision and comprehension answer to a specific rule, clause, and evidentiary or source field. Since operational bank data cannot be accessed for confidentiality and regulatory reasons, evaluation is performed on a synthetic, rule-grounded benchmark of 1,393 cases drawn from 215 rules. On the 294 held-out test cases, whose rules were never seen in training or tuning, the adaptive router reaches 92.2% automated accuracy and 96.9% end-to-end accuracy, compared with 85.4% for a fixed hybrid router, 72.5% for retrieval-augmented LLM-only reasoning, and 56.8% for LLM-only reasoning, and ablations show that removing adaptive routing, the rule engine, or evidence grounding lowers accuracy. Adversarial and ambiguous cases remain the hardest. All results are on synthetic rather than operational bank data, the decisions of the LLM reasoning paths in these runs were produced by a deterministic surrogate of the LLM judge, and results of the regulation-comprehension benchmark are not reported in this paper.**

***Index Terms*—RegTech, RBI regulations, regulatory intelligence, regulatory natural language processing, automated compliance checking, neuro-symbolic AI, retrieval-augmented generation, large language models, adaptive reasoning, banking supervision**

I. Introduction

Banking regulators like the Reserve Bank of India publish and constantly update a sprawling collection of Master Directions, circulars, notifications, and frequently-asked-question clarifications that in total define compliant conduct for scheduled commercial banks and non-banking financial companies. Two disparate classes of readers interact with this corpus, each of which deals with a different class of obstacles. Bank employees, compliance officers, and internal auditors alike must identify the one requirement of interest, correctly interpret the dense legal/regulatory language, and reconcile differences across multiple superseding edits—a task which is slow, error-prone, and not scalable given the pace at which the regulations themselves are updated. On the other hand, the regulator must supervise a large number of institutions, each

of which has to periodically submit audit reports, compliance reports and records, and has to decide if the evidence submitted by the institution actually meets the applicable requirement, a task which cannot be uniformly automated as the requirements are heterogeneous.

That heterogeneity is precisely the paper's starting observation. RBI regulatory text mixes numeric thresholds and ratios (for example, exposure limits expressed as a percentage of eligible capital), Boolean and procedural conditions (for example, the existence of a board resolution), temporal obligations (for example, a report that must be filed within a stated number of days), and qualitative standards phrased in natural language (for example, "reasonable measures" or "adequate due diligence") whose satisfaction cannot be reduced to a closed-form numeric test. A system that treats all of these uniformly misallocates reasoning effort in one of two directions. Delegating every check to a large language model inherits hallucination risk precisely on the numeric, high-stakes checks where a deterministic answer is both achievable and necessary. Conversely, restricting the system to hand-authored deterministic rules leaves the qualitative and context-dependent share of RBI regulatory text—which is substantial—entirely unaddressed.

This paper describes two modules, each of which targets one of the problems identified for the two different reader audiences. The Regulation Comprehension module targets the human reader: it processes an RBI circular into a structured, simplified, and searchable form amenable to answering natural language questions posed by a bank employee or auditor. Given a question, it returns an answer with justification in the form of a particular clause in the circular being answered, along with the source document.

The second module, Adaptive Compliance Checking, targets the machine reader: it consumes the same processed regulatory information to evaluate evidence submitted by the banks under audit. In this paper, we focus more on the second task. For this task, the technical challenge posed is the following: for a given regulatory requirement and the evidence about a bank's practices, what reasoning mechanism should be used to determine whether the evidence satisfies the requirement, and how should this choice be made in a principled way that does not rely on an ad-hoc confidence threshold? We call this problem reasoning-modality routing. We note that while routing between different language models has been widely studied—especially in the context of higher versus lower capability and cost models—our task is different in that we are not choosing between multiple language models. Rather, we are choosing between qualitatively different reasoning mechanisms, namely, a rule-based deterministic reasoner, a retrieval-augmented language model, and human intervention, for a single underlying language model.

A. Contributions

This paper makes the following contributions, ordered by the primacy assigned to them in Section XVIII.

- An adaptive, cost-aware reasoning-modality router for regulatory compliance checking, viewed as a constrained optimization over the rule-engine, retrieval-augmented-LLM, and human-review paths and evaluated on an accuracy–cost frontier against the static, confidence-threshold, and fixed-hybrid routing baselines (primary contribution).
- A schema-constrained rule-extraction pipeline, situated as part of a larger Regulation Comprehension module, that translates RBI Master Direction and circular clauses into a typed, composable, executable rule representation capturing the quantitative, Boolean, conditional, temporal, qualitative, relational, and multi-entity requirement categories, with per-category extraction accuracy defined as an evaluation target of the comprehension benchmark (secondary contribution).
- A systematically built, rule-based synthetic compliance benchmark covering the nine case-difficulty categories (compliant, non-compliant, boundary, near-boundary, ambiguous, missing-data, adversarial, contradictory, and cross-version) to address the lack of a public RBI-specific compliance benchmark (secondary contribution).
- A dedicated regulation-comprehension evaluation methodology—extraction, question-answering, and simplification sub-benchmarks, each with its own baselines and metrics (Sections XIII, XIV.B, XV-B)—designed to evaluate Module 1 with the same rigor as Module 2 rather than assuming that Module 1 works because Module 2 works (secondary contribution).
- An evidence-grounded decision architecture in which each compliance decision, and each answer returned by the regulation-comprehension question-answering interface, cites a rule identifier, regulatory clause, and evidentiary or source input (supporting contribution).
- A system-level integration in which the structured regulatory knowledge produced by Regulation Comprehension is directly reused by Adaptive Compliance Checking, so that a single pipeline forms the basis for both applications rather than two separate ones (system-level contribution).

B. What this paper does not claim

Consistent with the novelty analysis in Section XVIII, this paper does not claim that adding rule-engine and retrieval-augmented LLM reasoning is itself novel. Such hybrid architectures exist in recent regulatory-compliance natural language processing work (Section III). It does not claim that rule extraction from regulatory text is a new technique; that technique has a well-established lineage in the automated compliance checking literature for construction codes, applied here to a new domain. It does not claim as novel, in themselves, the techniques of retrieval-augmented generation, large language model use, or generic routing between language models. It does not claim validated connected-counterparty graph reasoning for exposure aggregation: category G is evaluated only on a small number of synthetic cases (Section XVI), because a meaningful evaluation would require multi-hop

ownership data that is not publicly available, and a fuller treatment is left to Section XX. It does not implement optical character recognition or scanned-document ingestion; bank-evidence inputs are restricted to structured formats and PDFs with extractable text. It does not report measured results for the Regulation Comprehension benchmark (Section XVI).

C. Paper organization

Section II formally defines the two-part research problem and states the research questions. Section III reviews the related work and defines the particular gap addressed by this work. Section IV describes the overall architecture of the integrated system. Section V fully describes the Regulation Comprehension module in technical detail. Section VI describes the representation of regulatory rules. Section VII describes the Adaptive Compliance Checking module in similar detail. The Adaptive Reasoning-Modality Router is described in Section VIII. Evidence-based explainability is discussed in Section IX. Section X describes the end-to-end integration of the two modules. Section XI discusses the implementation. Section XII describes the methodology for evaluating Adaptive Compliance Checking, and Section XIII similarly describes the methodology for evaluating Regulation Comprehension. Section XIV provides the baselines for both modules, and the evaluation metrics are described in Section XV (for Compliance Checking) and Section XV-B (for Regulation Comprehension). Section XVI reports the experimental results and their discussion. Section XVII describes the ablation studies for both modules. Section XVIII evaluates the novelty of this work. The limitations and threats to validity are discussed in Section XIX, and future work in Section XX. Finally, the conclusion is given in Section XXI.

II. Problem Statement and Research Objectives

A. Regulation comprehension

Let D be an RBI regulatory document (a Master Direction, circular, or notification). The regulation comprehension task is to construct a structured representation of D—clause segments, definitions, applicability conditions, temporal obligations, exceptions, and category labels—plus a simplified natural-language rendering suitable for a non-legal reader, and to enable natural-language question answering over D such that every answer cites the specific clause and source document that supports it.

B. Regulatory compliance verification

Let $R = \{r_1, \ldots, r_n\}$ denote the set of regulatory rules extracted from an RBI regulatory corpus (Section VI), where each rule r_i is represented as the typed object defined in Section VI. Let E denote a bank entity's evidence set: the structured and unstructured data available about that entity relevant to a given rule. A compliance query is a pair (r_i, E). The compliance-checking task is to produce a decision

$$d \in \{\text{COMPLIANT},\ \text{NON_COMPLIANT},\ \text{INSUFFICIENT_DATA},\ \text{AMBIGUOUS},\ \text{REQUIRES_HUMAN_REVIEW}\}, \quad (1)$$

together with an evidence trace τ identifying the specific rule identifier, clause reference, and evidence fields that justify d.

C. Research objectives and research questions

The general goal is to establish whether any single reasoning mechanism is universally optimal for checking compliance with RBI regulatory requirements, and whether a combination of a deterministic rule engine, a retrieval-grounded LLM, and escalation to humans would allow attaining higher compliance accuracy than any single mechanism used alone, with similar or better cost and latency. The research questions are defined below.

- **RQ1 (Compliance accuracy):** Is adaptive reasoning-modality routing beneficial compared to rule-only, LLM-only, retrieval-augmented-LLM-only, and fixed-hybrid baselines in terms of overall and per-case-type compliance accuracy?
- **RQ2 (Routing efficiency):** Can the proposed routing algorithm produce a better accuracy–cost–latency trade-off than static, confidence-threshold, and fixed-hybrid routing strategies?
- **RQ3 (Rule and regulation extraction quality):** What is the precision, recall, and executability of rules and structured regulatory knowledge extracted from RBI text, grouped by requirement category?
- **RQ4 (Evidence grounding and hallucination):** What is the reduction in hallucinated or unsupported output across different types of cases, in the regulation-comprehension question-answering interface and the compliance decision architecture, compared to an ungrounded LLM baseline?
- **RQ5 (Robustness to difficult cases):** What is the degradation in performance across boundary, near-boundary, ambiguous, missing-data, adversarial, contradictory, and cross-version cases compared to compliant and non-compliant cases?
- **RQ6 (Regulation comprehension quality):** How accurate is the retrieval of the clause(s) relevant to a natural-language regulatory question, and how faithful and relevant are the generated answers, compared to keyword-search, LLM-only, and generic-RAG baselines?
- **RQ7 (Simplification fidelity):** Does simplification of RBI-specific clauses lead to a significant improvement in readability while maintaining the fidelity of stated conditions and exceptions compared to the source clause?

III. Related Work

A. Automated compliance checking and rule extraction from regulatory text

Automated compliance checking has a substantial research history predating large language models, focused on the construction and building-code domain. Zhang and El-Gohary [1] propose a semantic, pattern-based natural language processing approach to extracting information-extraction and conflict-resolution rules from construction regulatory documents, decreasing the number of hand-authored extraction patterns needed by using syntactic and semantic text features. A related

line of work employs ontology- and rule-based natural language processing to interpret utility-infrastructure regulations, decomposing text into requirement, applicability, selection, and exception components [2]. More recently, Hettiarachchi et al. [3] introduce a manually annotated corpus of building-regulation sentences with entity and relation labels to aid machine-readable rule generation with modern natural language processing and large language model methods, motivating a shift from hand-authored extraction patterns toward learned extraction as regulatory domains multiply. This line of work establishes the feasibility and the standard evaluation methodology—entity, operator, and threshold precision/recall/F1, together with rule executability—for extracting structured rules from regulatory natural language, a methodology we adopt and adapt to RBI banking regulation in Section V without claiming to have originated it.

B. Neuro-symbolic and hybrid regulatory compliance systems

A related and more recent line of work applies hybrid neural and symbolic architectures directly to financial and regulatory compliance. Ershov [4] demonstrates construction of an executable regulatory knowledge graph from a BERT-tagged corpus of Abu Dhabi Global Market regulations using coreference resolution and syntactic analysis to populate a graph database queried via a graph query language, and proposes graph neural networks as a direction for future judgment automation. Subsequent work explicitly frames compliance reasoning as knowledge-graph-augmented retrieval-augmented generation, monitoring of regulatory change, extraction of structured obligations, and detection of compliance gaps against internal policy jointly across multiple regulatory frameworks, and reports grounding-accuracy figures as a means of quantifying hallucination mitigation from knowledge-graph grounding [5]. Independently, symbolic reasoning layers (SMT solvers) over constraints extracted by LLMs from financial regulations have been proposed for compliance analysis [6], and graph-neural-network-based transaction monitoring coupled with an LLM explanation layer has been evaluated on simulated transaction streams for compliance flagging [7]. Code generation as an intermediate, executable representation for financial compliance checking has been explored [8] as well. Collectively, this literature establishes that grounding LLM compliance reasoning in an extracted symbolic representation is an active area of research, instantiated in numerous ways, which substantially narrows the novelty available to a system that merely combines these components (Section XVIII).

C. Regulatory question answering and regulation comprehension

In addition to regulatory decisioning, another related strand of work is concerned with the human-facing question of comprehending regulatory text: distilling dense legal or regulatory language into a form amenable to query and interpretation by a non-specialist reader. Retrieval-augmented generation—a generator conditioned on passages retrieved from an external corpus, as opposed to parametric knowledge alone—was introduced by Lewis et al. [9] as a general-purpose mechanism for grounding generation in retrievable evidence, and is the dominant architectural pattern for regulatory and legal question answering in the broader literature (Section III-E). Feeding this pipeline requires first recovering a document's textual and structural content faithfully; layout-aware document understanding models such as LayoutLM [10], which jointly model text and layout signals rather than treating a document as a flat token stream, establish that structural cues (headings, paragraph numbering, tabular layout) materially improve downstream extraction from regulatory-style documents, motivating the layout-aware ingestion step of Regulation Comprehension (Section V).

A retrieval-augmented generator is not guaranteed to stay faithful to what was retrieved: a further strand of work is concerned with evaluating and mitigating this gap. Reference-free RAG evaluation frameworks such as RAGAs [11] decompose RAG quality into retrieval-side metrics (context precision, context recall) and generation-side metrics (faithfulness, answer relevance), which we adopt as the basis for the comprehension-side evaluation methodology in Section XIII; and the broader survey of hallucination in natural language generation by Ji et al. [12] catalogs the mitigation strategies—of which citation-constrained generation and retrieval grounding are two—that motivate the hallucination-mitigation design of Section V. The Regulation Comprehension section of this work is built using the retrieval-plus-citation paradigm described above, but applied to RBI Master Directions and circulars. In particular, we are not aware of any published system that handles RBI documents' category structure (KYC, anti-money laundering, lending, transaction reporting, governance, audit), their amendment relationships, and effective-date semantics as described in this paper. Legal contract understanding datasets such as CUAD [13]—where spans of text from over 500 annotated commercial contracts are tasked to models as span-based QA to retrieve contract clauses—and financial QA benchmarks such as FinQA [14]—which require reasoning over tables and text to perform multi-step arithmetic operations—define the general approach of QA over domain-specific text with spans or programs as possible answers and expert annotations as ground truth, which is adapted in Section XIII for the task of RBI regulation comprehension, but without targeting banking regulations in particular.

D. RBI-specific and Indian banking compliance automation

Commercial governance-risk-compliance platforms promoting automated RBI compliance offer products for regulatory mapping and automated reporting, but describe product capability rather than reproducible research methodology, benchmarks, or evaluation [15]. We were not able to locate a peer-reviewed system targeting RBI-specific regulations, evaluating rule or regulation extraction accuracy against RBI's clause structure across Master Directions, circulars, and FAQs (including amendment and supersession relations), or publishing a benchmark for RBI compliance reasoning. We treat this

as the primary domain-application gap addressed by this paper, while remaining explicit that the underlying techniques (Sections III.A–B) are not new.

E. Retrieval-augmented generation and LLM-based regulatory reasoning

The use of retrieval-augmented generation to condition a language model's output on passages retrieved from an external corpus, rather than purely parametric knowledge, is now the standard approach to ground LLM outputs in source text in the legal, financial, and regulatory domains, and underlies both the question-answering component of Regulation Comprehension (Section V) and the qualitative-reasoning path of Adaptive Compliance Checking (Section VII). We do not claim retrieval-augmented generation itself as a contribution; our use of it is standard, and the paper's contribution lies instead in when the system chooses to invoke it relative to deterministic reasoning (Section VIII).

F. LLM routing

A burgeoning literature focuses on routing a query to the most cost-efficient sufficiently-capable large language model (LLM) from a pool of options. Hybrid LLM [16] and BEST-Route [17] operationalize quality–cost trade-offs; RouteLLM [18] learns routing policies from preference data; GraphRouter [19] and universal model routing [20] develop joint embeddings of models and queries to enable learned selection; cascade- and uncertainty-based routers route to a cheap model by default and defer to expensive ones when uncertain [21]; and survey and benchmarking papers provide standardized evaluation protocols for the task [22]. This literature uniformly focuses on selecting from different LLMs (of varying capability and cost) for a given prompting strategy. By contrast, our routing problem (Section VIII) concerns choosing between qualitatively different reasoning modalities for a fixed prompting strategy. We aim to route to whichever of a symbolic reasoner, retrieval-augmented LLM, or human can answer a question with a defensible correctness guarantee, for a given LLM (not across LLMs). We adopt the evaluation paradigm of the literature, comparing performance on an accuracy–cost–latency frontier against static, threshold, and learned baselines, rather than reporting one aggregate accuracy metric (see Sections XII–XVI), while noting that these works do not consider this particular routing task for compliance reasoning.

G. Synthetic benchmarks for financial and regulatory AI

Recent work argues that static question-answer benchmarks are insufficient validation for financial and compliance-oriented LLM systems, and explicitly calls for benchmarks incorporating retrieval evidence requirements, escalation scenarios, and adversarial cases [23]. Logic-guided synthesis has been used to construct test cases probing implicit regulatory-compliance violations in LLM tool invocation across financial, healthcare, and Internet-of-Things domains [24], and semi-synthetic benchmark generation has been proposed specifically to produce contamination-resistant financial evaluation data [25]. Our benchmark construction methodology (Section XII) follows this emerging practice—generating test cases systematically from extracted rules rather than independently—and explicitly includes boundary, adversarial, missing-data, and cross-version cases, although, to our knowledge, it is among the first such benchmarks grounded specifically in RBI regulatory text. Our extraction and retrieval components also build on general legal natural-language-processing foundations, including domain-adapted language models [26] and benchmark suites for legal reasoning and legal language understanding [27], [28], which establish that general-purpose language understanding transfers imperfectly to legal and regulatory text.

H. Research gap

No prior system, to the best of our knowledge, (a) formalizes and evaluates routing between deterministic and neural reasoning modalities (rather than between neural models) for compliance checking; (b) applies rule extraction and hybrid reasoning explicitly to RBI regulatory text with a reported breakdown of per-category accuracy, jointly with a human-facing regulation comprehension interface built from said extracted representation; or (c) provides a rule-grounded synthetic benchmark for Indian banking regulatory compliance intended for public release. This paper addresses these gaps through the system design, the benchmark construction, and an experimental evaluation of the compliance-checking module, with (a) being our primary contribution. Table I summarizes the relation between our proposed system and the categories of prior work identified above.

IV. System Overview

The system comprises two interconnected modules, as opposed to two standalone applications. Regulation Comprehension (Section V) transforms an RBI regulatory document into structured, machine-readable regulatory knowledge and a human-facing question-answering interface. Adaptive Compliance Checking (Section VII) consumes the same structured knowledge to evaluate bank-submitted evidence from RBI's supervisory perspective. The two modules share a single regulatory knowledge base, as opposed to maintaining separate representations, which is the basis of the integration contribution discussed in Section X.

The entire pipeline, summarized in Table II and illustrated for its two modules in Fig. 1 and Fig. 2, works as follows: an RBI circular or master direction is input and processed; regulation comprehension extracts relevant regulatory knowledge from it, which is then used both to populate a regulatory search and question-answering interface for human reviewers and to build a compliance rule repository for the adaptive compliance checking component; bank evidence is input and normalized; the adaptive compliance router selects, for each rule and each evidence set, whether the deterministic rule

TABLE I
COMPARISON WITH CATEGORIES OF EXISTING APPROACHES DISCUSSED IN SECTION III.

Approach	Regulation Understanding	Rule-Based Verification	LLM Reasoning	Adaptive Modality Routing	RBI-Specific	Human Escalation	Evidence Traceability
Construction-code ACC [1]–[3]	Partial	Yes	No	No	No	No	Partial
Knowledge-graph RAG compliance [4]–[7]	Yes	Partial	Yes	No	No	No	Yes
Compliance-to-Code [8]	Partial	Yes	Yes	No	No	No	Partial
Commercial RBI GRC platforms [15]	Yes	Yes	Unknown	Unknown	Yes	Unknown	Unknown
LLM-to-LLM routing [16]–[22]	No	No	Yes	No (routes models, not modalities)	No	No	No
Layout-aware document QA [9], [10], [13], [14]	Yes	No	Yes	No	No	No	Partial
Proposed system	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

engine, the retrieval-augmented large language model, or human judgment should be used to reach a compliance decision; evidence is aggregated and a decision is made; and finally, a supervisory audit report is generated with citations of the relevant rule, clause, evidence, the method of reasoning used, and recommended remedial action, if any. Table II summarizes the processing layers and the purpose of each.

A. Scope decisions

Two decisions in this paper, material to the implementation and evaluation scope, are articulated explicitly rather than assumed. First, connected-counterparty graph reasoning for large-exposure aggregation across related entities is evaluated only at a limited scale (category G, 8 test cases; Section XVI), since its meaningful evaluation would require multi-hop ownership data not available in the public domain, so that any statement of accuracy reliant on synthetic ownership data generated by the authors is a validation loop too weak to support a general claim; a fuller treatment is discussed in Section XX as future work. Second, implementation of optical character recognition and scanned-document ingestion is out of scope for the evaluated system: both regulatory-document and bank-evidence inputs are constrained to formats supporting extractable text (structured CSV/JSON, and text-layer PDFs), since OCR quality is an orthogonal engineering concern not germane to the research questions of Section II.

V. REGULATION COMPREHENSION MODULE

This section describes the module that turns an RBI regulatory document into structured knowledge and a human-facing comprehension interface, and whose output is reused directly by Adaptive Compliance Checking (Section X). We describe this module in the same technical depth as Adaptive Compliance Checking (Section VII) and define a dedicated evaluation methodology, metrics, and baselines for it (Section XIII), rather than treating it as a preprocessing step subordinate to compliance checking. The evaluation design measures the two modules independently because they answer different questions—"what does the regulation say and mean" versus "does this bank satisfy it"—and a system that comprehends regulation well but checks compliance poorly, or vice versa, is a materially different (and differently deployable) system from one that does both well; collapsing the two into a single reported number would hide that distinction.

A. RBI document ingestion

The module accepts an RBI circular or Master Direction in the form of a PDF, along with associated metadata (PDF title, issue date, document type (Master Direction, Master Circular, or standalone circular), and, if present, the department of RBI that issued it (e.g., the Department of Regulation or the Department of Banking Supervision)). Ingestion is idempotent (resubmitting an ingested document is detected by content hash and results in no duplicate entry in the knowledge base), which is important operationally, as RBI frequently republishes a document with only its "updated as on" date changed following a minor amendment.

B. PDF/text extraction

Layout-aware text extraction recovers the text of the document while preserving paragraph and sub-paragraph numbering, headings, and table layout, as RBI documents use consistent structural numbering to indicate clause hierarchy. We take as given the finding of Xu et al. [10] for layout-aware document pre-training that treating a regulatory document as a flat token stream and throwing away positional and typographic structure demonstrably degrades downstream extraction; the extraction step therefore preserves each text span's page position, indentation level, and any enclosing table cell coordinates, rather than emitting a single concatenated string. RBI Master Directions tend to nest content four levels deep—Chapter, Part, Paragraph, and sub-paragraph (e.g., "Chapter III, Para 7, sub-para (3)(ii)")—and the extraction step reconstructs this nesting explicitly rather than inferring it later from indentation heuristics applied to already-flattened text.

C. Document cleaning and normalization

The extracted text is normalized in order to remove page-header and page-footer artifacts (RBI documents place a running header with the circular reference number and a footer with the page number and department name on every page, both of which would otherwise be mis-segmented as body content), to repair hyphenation introduced by line wrapping, and to standardize whitespace, without altering the substantive wording of the regulatory text. Footnote markers and their corresponding footnote text (common in RBI Master Directions for citing an amending circular) are detected and re-attached to the clause they annotate rather than left as an orphaned trailing block.

TABLE II
SYSTEM PROCESSING LAYERS.

Layer	Function	Module	Processing stage
1	Regulatory document ingestion (PDF/HTML acquisition, cleaning, section identification)	Regulation Comprehension	Offline
2	Clause segmentation, definition extraction, cross-reference and effective-date detection	Regulation Comprehension	Offline
3	Rule extraction and classification into the typed schema (Section VI)	Regulation Comprehension	Offline
4a	Relational rule store: structured rules, versions, applicability, thresholds	Shared knowledge base	Offline
4b	Vector store: clause and definition embeddings for semantic retrieval	Shared knowledge base	Offline
5	Regulatory search / question answering with clause-level citation	Regulation Comprehension	Query-time
6	Bank evidence ingestion and canonicalization into structured evidence fields	Adaptive Compliance Checking	Query-time
7	Adaptive reasoning-modality router (Section VIII)	Adaptive Compliance Checking	Query-time
8	Reasoning: deterministic rule execution or retrieval-grounded LLM reasoning	Adaptive Compliance Checking	Query-time
9	Evidence aggregation, conflict detection, and compliance decision assignment	Adaptive Compliance Checking	Query-time
10	Explainability: audit-ready supervisory report generation	Adaptive Compliance Checking	Query-time

D. Clause segmentation

Clause segmentation is determined deterministically based on RBI's regular structural markers (paragraph and sub-paragraph numbering conventions in a given document type) rather than by an LLM, as these are sufficiently reliable for deterministic, rule-based parsing and hence save model calls for tasks that actually require semantic understanding. A segmentation rule set (regular-expression patterns matched against the numbering conventions of Master Directions, Master Circulars, and standalone circulars separately, as the three document types number differently) yields a candidate set of clause boundaries that are validated against the document's own table of contents (if any) as a consistency check: a document whose inferred count of clauses differs by more than a small tolerance from its table of contents is flagged for manual review rather than silently ingested with a possibly incorrect clause structure.

E. Regulatory definition extraction

Defined terms (e.g., a Master Direction's "Definitions" clause, typically numbered as an early paragraph such as "Para 3 – Definitions") are extracted and linked to every subsequent clause in which the term is used, so that downstream simplification and question answering can resolve a term to its governing definition rather than relying on the reader's prior familiarity with it. Term linking uses exact and near-exact string matching against the extracted definition list within the scope of the document (and, where cross-reference handling below has resolved a link, within cross-referenced documents), rather than a general-purpose named-entity recognizer, because RBI definitions are self-contained and enumerated rather than requiring open-domain entity recognition.

F. Rule and obligation identification

Each segmented clause is analysed to determine whether it imposes an obligation ("banks shall ..."), a prohibition ("no bank shall ..."), a permission ("banks may ..."), or is a definition or purely explanatory or preambular statement; the rule-extraction pipeline described in Section VI builds on this labeling. For this purpose, an LLM is prompted with a small set of in-context examples covering each of the five categories together with the deontic-verb patterns (shall, shall not, may, is required to, is exempt from) RBI text uses to express them, with its output constrained to one of the five labels rather than free text.

G. Regulatory category classification

Clauses are categorized under operational categories such as Know Your Customer (KYC), anti-money-laundering (AML), lending, transaction reporting, governance, and audit, so that both the human-facing search interface and the compliance rule repository can be filtered by category. A clause may receive multiple category labels (for example, a reporting obligation attached to a KYC requirement is labeled both KYC and transaction reporting), because RBI requirements often span multiple operational areas and assigning a single label would lose that overlap.

H. Simplification of regulatory language

For each clause, the module produces a simplified natural-language restatement for the benefit of a non-legal reader, created under an LLM prompting strategy which favors preserving the clause's substantive conditions and obligations over simply shortening the wording; the original clause text is the authoritative source and is always shown in conjunction with the simplification, and the interface never presents the simplification as a replacement the user cannot trace back to source. Quality of the simplification is assessed along two axes that can trade off against each other—how much simpler the text has become, and how much of the original meaning has survived—using the metrics discussed in Section XV-B, item C, since a simplification that is easy to read but drops an exception clause is not actually helpful to a compliance officer.

I. Structured regulatory knowledge representation

The output of clause-level processing is stored as the typed rule representation described in Section VI, which captures quantitative, Boolean, conditional, temporal, qualitative, relational, and multi-entity content within a single schema, such that regulation comprehension and compliance checking are read from the same object instead of two independently maintained representations of the same clause.

J. Cross-reference handling

References from one clause to another ("as specified in paragraph 4 above") or to a separate Master Direction or circular ("in terms of the Master Direction on KYC dated . . . ") are resolved to the target clause identifier where the target has itself been ingested, so that a rule's applicability or exception conditions can be traced across documents rather than treated as opaque text. Where the referenced document has not yet been ingested, the reference is retained as an unresolved pointer (source document name and, where extractable, a date) rather than silently discarded, so that it can be resolved automatically once that document is ingested.

K. Amendment and supersession handling

When a newly ingested document amends or supersedes a previously ingested clause, the module records a supersession relationship (Section VI, field `supersedes`) rather than silently overwriting the prior version, as per the versioning approach described below. Since RBI amendments are frequently partial—a circular may amend a single subparagraph of a much longer Master Direction—the module records supersession at the clause level, not the document level, so that the ninety-nine unamended clauses of a Master Direction are not needlessly revisioned when only one clause changes.

L. Version and effective-date management

Every extracted rule records an effective-from date and, if relevant, an effective-to date, so that a query targeted at a point in time can be resolved against the regulation as it was at that date and not just the most up-to-date version. This is what makes the cross-version benchmark cases of Section XII meaningful: a compliance query dated before an amendment's effective date must be answered according to the pre-amendment rule, and the same evidence evaluated after that date must be answered according to the amended rule, with the same underlying versioning metadata rather than a special-cased comparison.

M. Retrieval mechanism

Clause and definition text is embedded and indexed for semantic retrieval, such that a natural-language query can be matched to the most relevant clauses even when the query's wording does not lexically match the regulatory text—e.g., a user asking about "cash transaction limits" should be able to retrieve a clause phrased in terms of "currency transaction reports" even if the two have no keywords in common. Retrieval quality is measured directly (Section XIII.B) rather than assumed, as a citation-constrained answer generator (below) can only be as accurate as the clauses it is given to work with.

N. Natural-language question answering

A user question is answered by retrieving the most relevant clauses, followed by generation of a response conditioned on the retrieved text under a citation-constrained prompting strategy, following the retrieval-augmented generation pattern introduced by Lewis et al. [9] and discussed further in Section III-C. The generator is instructed to answer using only the retrieved clauses, to quote or closely paraphrase the operative condition rather than the surrounding preamble, and to state explicitly when the retrieved clauses do not fully answer the question rather than filling the gap from unretrieved background knowledge. Fig. 1 shows the question-answering workflow of the module, from query analysis through retrieval, fusion, and grounded answer generation to the cited sources and the plain-language restatement.

O. Source and clause citation

Every answer returned by the question-answering interface explicitly cites the RBI circular, the clause identifier and, if applicable, the regulation version, allowing the user to verify against the source rather than against the system's restatement. This citation is stored as structured metadata (document id, clause id, version) together with the generated text, and is not inferred a posteriori by pattern-matching the answers against the corpus, since that would only find citation errors rather than prevent them.

P. Hallucination mitigation

Beyond citations, the module is limited to the retrieved passages for answering, and refrains from speculating on queries for which the retrieved contextual evidence was found to be insufficient with respect to the target answer, thus avoiding hallucination; this is the same evidence-grounding design choice as for compliance decisions in Section IX, applied to the human-facing responses. We use the faithfulness and answer-relevance decomposition from the RAGAs evaluation framework [11] as a basis for our own metric for this property (Section XIII.C). The broader hallucination-mitigation taxonomy of Ji et al. [12], of which our grounding, citation, and abstention mechanisms are specific examples, informs our approach to this design choice, rather than retrieved data being considered inherently safer than generation.

Q. Reuse by the compliance checking module

The structured regulatory knowledge produced by this module—typed rule objects, clause citations, category labels, and version and effective-date metadata—is not re-generated for Adaptive Compliance Checking; it is read directly from the shared regulatory knowledge base described in Section X. This reuse is the basis of the paper's contribution at the system level.

R. Worked example

To make the module's behavior concrete, consider RBI's Know Your Customer Directions for commercial banks, paragraph 41, which requires the bank to align the extent of monitoring with the risk category of the customer, to subject high-risk accounts to more intensified monitoring, and to establish the need for enhanced due diligence measures. Ingestion (Sections V.A–B) recovers the paragraph's text together

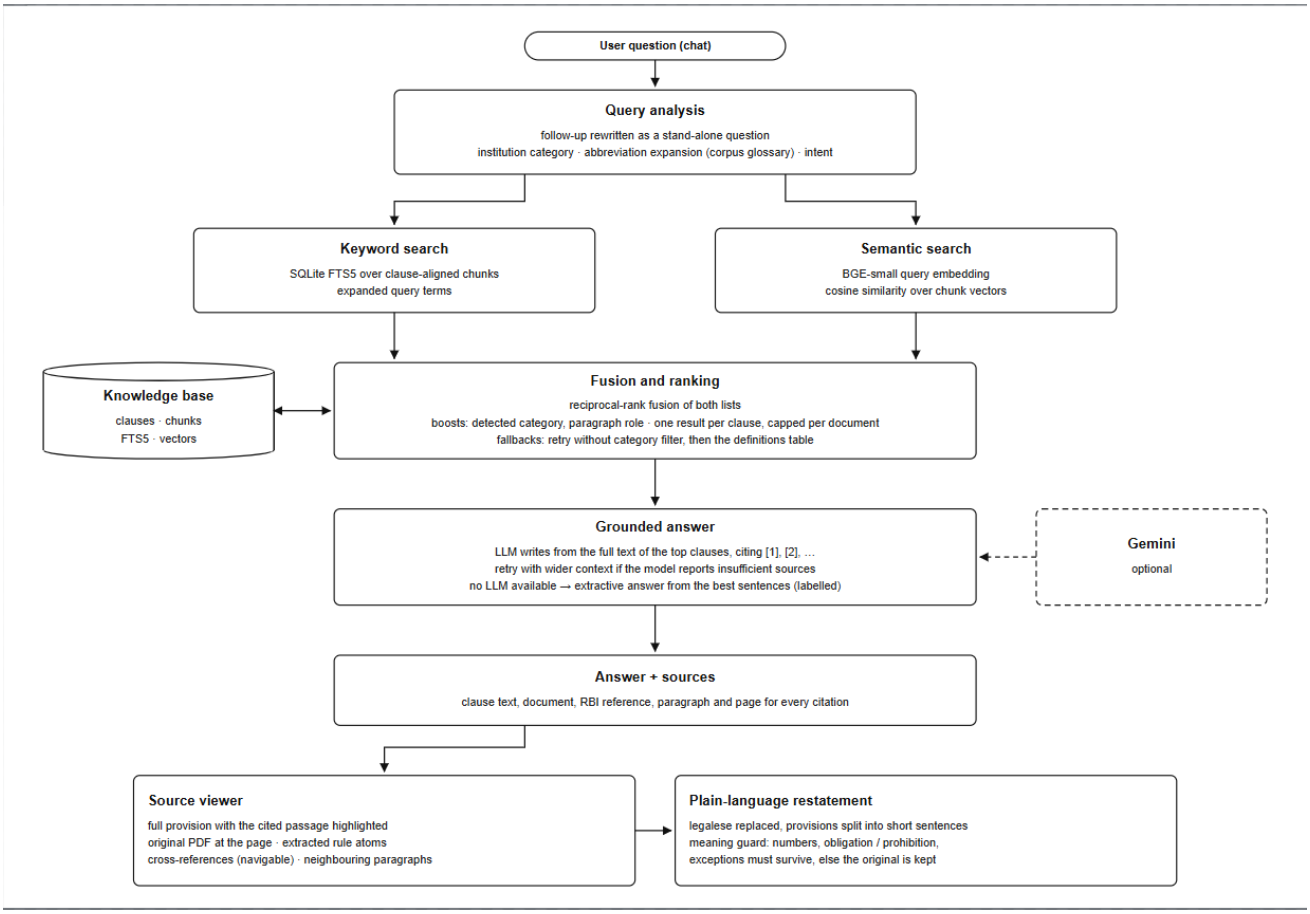


Fig. 1. Question-answering workflow of the Regulation Comprehension module: query analysis, keyword and semantic retrieval over clause-aligned chunks, rank fusion, grounded answer generation with clause-level citations (with an extractive fallback when no LLM is available), and a source viewer and plain-language restatement of the cited provisions.

with its position under Chapter VI (Customer Due Diligence Procedure); segmentation (Section V.D) isolates it as a single clause distinct from the surrounding paragraphs 40 and 42; definition linking (Section V.E) resolves the term "on-going due diligence" to its governing definition in paragraph 5; rule identification (Section V.F) labels the clause as an obligation; category classification (Section V.G) labels it KYC; and rule extraction (Section VI) produces a category-E (qualitative) rule object with `requires_semantic_interpretation = true`, since "enhanced due diligence" is not reducible to a numeric or Boolean test. A subsequent user question, "what extra checks are needed for a high-risk customer," retrieves this clause (Section V.M) and receives a simplified, citation-bearing answer (Sections V.I, N–O) referencing the KYC Directions, paragraph 41. The same rule object, unchanged, is later read by Adaptive Compliance Checking (Section VII) when evaluating a specific bank's due-diligence file for a high-risk customer, and because `requires_semantic_interpretation = true`, the router (Section VIII) directs that evaluation to the retrieval-augmented LLM path rather than the rule engine.

VI. Regulatory Rule Representation

A flat (condition, operator, threshold) triple is insufficient to express conditional ("if … and … then … unless"), temporal, or exception-bearing clauses without ad hoc extensions; instead, every extracted rule is represented as a typed object with a nested logic block, summarized in Table III.

A. Requirement category taxonomy

The extracted clauses were categorized into seven types, summarized in Table IV. Clauses of types A–D, along with the Boolean-evidence subset of type E and the non-graph cases of type G, were prioritized for deterministic rule-engine processing if applicable; the remaining cases of type E and the unresolved cases of type F were processed by retrieval-grounded LLM reasoning (Section VIII).

VII. Adaptive Compliance Checking Module

This module addresses the supervisory side of the problem: given a bank's submitted evidence and the applicable RBI regulatory requirements, it produces a traceable compliance decision. As with Regulation Comprehension (Section V), we describe each processing step in enough detail to support independent implementation and evaluation, since this

TABLE III
CORE FIELDS OF THE TYPED RULE REPRESENTATION (ABRIDGED).

Field	Type	Purpose
rule_id	string	Stable identifier, includes source document and clause
source_document / clause_id	string	Traceability to the originating Master Direction / circular
rule_type	enum (A–G)	Quantitative, Boolean, conditional, temporal, qualitative, relational, multi-entity
applicability	object	Entity types and conditions under which the rule applies
logic.antecedent / consequent / exceptions	nested object	Composable Boolean, conditional, and quantitative logic in one schema
temporal	object	effective_from, effective_to, reporting_frequency
requires_semantic_interpretation	boolean	Set at extraction time; primary router feature (Section VIII)
confidence / human_verified	float / boolean	Extraction-quality metadata, independent of downstream decision confidence
version / supersedes	int / rule_id	Versioning and amendment tracking (Section V)

TABLE IV
REQUIREMENT CATEGORY TAXONOMY AND EVALUATED-SCOPE STATUS.

Category	Description	Preferred reasoning modality	Status
A – Quantitative	Thresholds, ratios, numerical limits	Rule engine (evidence complete)	Implemented / evaluated
B – Boolean / procedural	Existence checks (e.g., board approval, documentation)	Rule engine (evidence complete)	Implemented / evaluated
C – Conditional	If … and … then … unless clauses	Rule engine (evidence complete)	Implemented / evaluated
D – Temporal	Deadlines, periodic reporting, date-conditioned applicability	Rule engine (evidence complete)	Implemented / evaluated
E – Qualitative / semantic	"Adequate due diligence," "reasonable measures"	Retrieval-augmented LLM	Implemented / evaluated
F – Relational	Parent/subsidiary, related-party requirements	Retrieval-augmented LLM (non-graph cases)	Partially implemented
G – Multi-entity / connected-counterparty	Exposure aggregation across related entities	Rule engine / graph check	Limited synthetic evaluation (Section XVI); fuller treatment future work (Section XX)

module—specifically its routing component (Section VIII)—carries the paper's primary contribution and is evaluated on the benchmark of Section XII. Fig. 2 gives an overview of the module's workflow, from the automatic check of a public document and the exploration of stored bank data, through rule applicability and the adaptive router, to the decision layer and the compliance report.

A. Bank data ingestion

Banks periodically submit audit reports, compliance filings, and operational records in structured (CSV/JSON) or text-extractable PDF form, per the scope decision in Section IV.A. A submission is tagged with the submitting entity's identifier, the reporting period it covers, and the RBI return or report type it corresponds to (for example, a DSB return or an internal audit report), so that evidence from different reporting periods for the same entity is never silently merged.

B. Data normalization

The submitted data is mapped to a canonical evidence schema, such that fields referenced by a rule's antecedent (e.g., "single-counterparty exposure as a fraction of eligible capital") can be found consistently independently of the format in which the bank originally reported that information. Since internal systems of different banks may export the same underlying quantity under different column names or units (e.g., exposure reported in absolute currency vs. already expressed as a ratio), the normalization step also performs unit reconciliation against the unit specified in the corresponding rule's schema (Section VI, field `unit`), flagging a mismatch instead of silently coercing it.

C. Evidence extraction

Where a submission is a text-extractable document rather than structured data, relevant evidence values and supporting statements are extracted and linked to the canonical schema field they populate, using the same layout-aware extraction approach used for regulatory documents (Section V.B), since bank audit reports share the same practical difficulty of mixing narrative text with embedded tables.

D. Evidence completeness

For each rule being evaluated, the module calculates the fraction of the rule's required evidence fields that are present and non-null. This fraction, called e-completeness, is one of the features of the router described in Section VIII. Completeness is calculated per rule, not per submission, since a single bank submission can be complete for one rule (and therefore sufficient evidence for it), but incomplete for another rule which requires an additional field not present in the submission.

E. Matching bank evidence with regulatory requirements

Applicable rules for a given bank and reporting period are selected from the shared regulatory knowledge base according to the rule's applicability conditions (bank entity type, bank category, and any additional qualifiers), read directly from the knowledge base populated by Regulation Comprehension (Section X) rather than a separate rule list.

F. Rule applicability

A rule is applicable to a given set of evidence only if the entity and conditions recorded in the applicability object are satisfied; inapplicable rules are thus excluded from the compliance evaluation for that entity (rather than being evaluated and then discarded, as would be the case if they were erroneously declared NON_COMPLIANT).

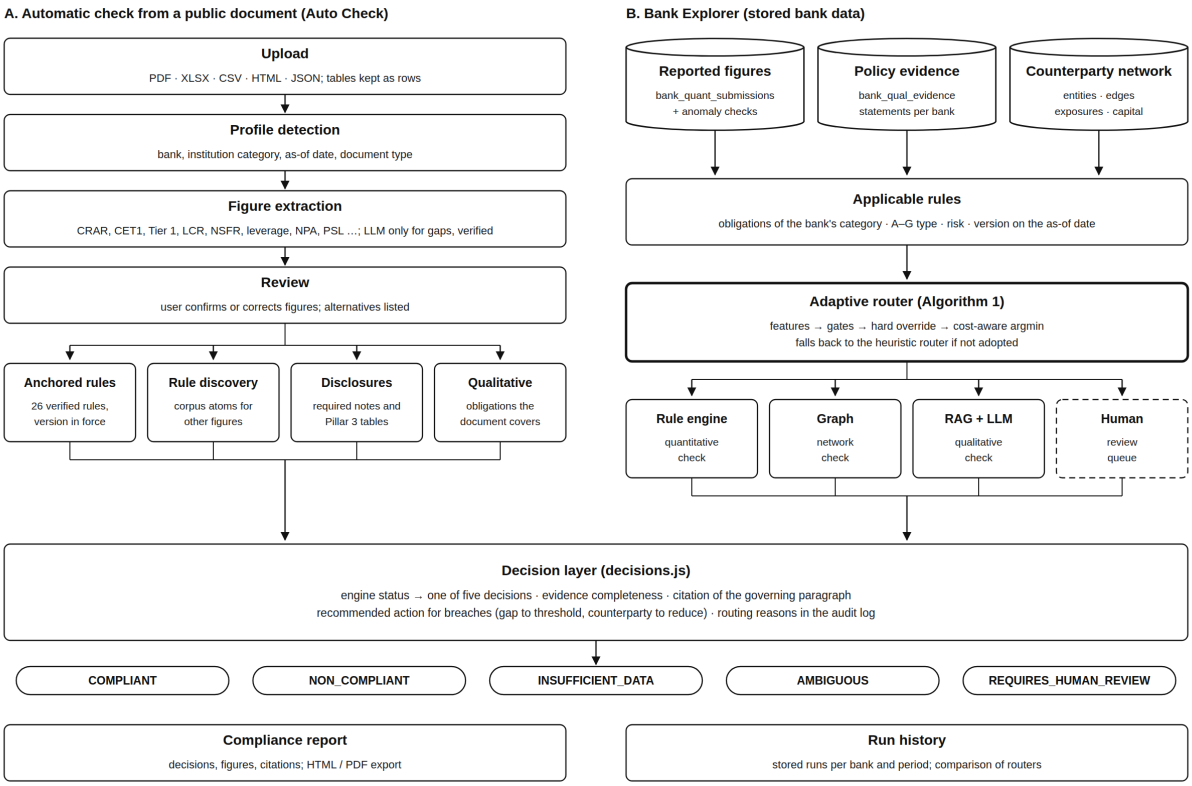


Fig. 2. Workflow of the Adaptive Compliance Checking module: (A) automatic check of a public document and (B) checking of stored bank data, in which applicable rules are passed to the adaptive router and dispatched to the rule engine, graph check, retrieval-augmented LLM, or human review, followed by a decision layer that assigns one of the five decisions and produces the compliance report. The graph-check branch corresponds to category G, which is evaluated only on a small synthetic subset (Section XVI).

G. Compliance decision

For each applicable rule, the routed reasoning modality (Section VIII) produces one of five decisions, summarized in Section VII.K below.

H. Evidence traceability

Every decision is stored together with the evidence trace τ defined in Section II, linking the decision to the specific rule, clause, and evidence fields that produced it (Section IX), so that a reviewer can inspect exactly which submitted values were consulted and which regulatory clause they were tested against.

I. Corrective recommendations

Where a decision is NON_COMPLIANT, the system generates a recommended corrective action derived from the rule's stated requirement (for example, "bring single-counterparty exposure within 20% of the eligible capital base or document an applicable exemption"), distinguished in the report from the compliance decision itself so that a recommendation is never presented as though it were a binding regulatory determination.

J. Audit-ready report generation

An audit-ready report is generated using a template that specifies the rule identifier, clause, evidence, reasoning modality used, decision, explanation, and any corrective recommendation, as described further in Section IX. The report format is kept consistent across different reasoning modalities, so that a reader cannot distinguish between a rule-engine decision and an LLM-produced one based on the report structure; the reasoning-modality field itself, not the report shape, is what indicates this.

K. Compliance decision outputs

The system returns one of five decisions for each (rule, evidence) query:

- **COMPLIANT**: the available evidence meets the rule's stated requirement.
- **NON_COMPLIANT**: the available evidence affirmatively fails the rule's stated requirement.
- **INSUFFICIENT_DATA**: required evidence fields are missing, and the system will not infer a decision from incomplete evidence.
- **AMBIGUOUS**: the available evidence neither meets nor violates the rule, most commonly for qualitative (category E) requirements, or if the sources of evidence disagree.
- **REQUIRES_HUMAN_REVIEW**: the router (Section VIII) has determined that no automated modality can provide a sufficiently defensible decision and the case is escalated.

L. Worked example

Continuing the example of Section V-R: suppose Bank X is submitting the due-diligence file for a customer classified, under the bank's risk categorisation, as high risk. Data normalization (Section VII.B) maps the fields of the submission into the canonical schema; evidence completeness (Section VII.D) is computed against the rule's category-E required fields (in this case, narrative fields describing the due-diligence steps taken, which do not reduce to a numeric completeness check as a quantitative rule's fields would); rule applicability (Section VII.F) confirms the rule applies to the entity type of Bank X; and because the rule's `requires_semantic_interpretation` flag is true (Section V-R), the router (Section VIII) sends the query to the retrieval-augmented LLM path rather than the rule engine, which reasons over the submitted due-diligence narrative against the retrieved clause text and returns COMPLIANT, AMBIGUOUS, or NON_COMPLIANT together with an evidence trace citing both the submitted narrative field and the KYC Directions, paragraph 41. A second, unrelated query for the same bank—whether its single-counterparty exposure evidence satisfies a quantitative large-exposure threshold—has `rule_engine_applicability = 1` and, if evidence is complete, is sent to the deterministic rule engine under the hard override of Section VIII.E regardless of the qualitative case just described, illustrating within a single bank's submission how the same pipeline routes two different requirement types to two different reasoning modalities.

VIII. Adaptive Reasoning-Modality Router

This section presents the paper's primary contribution: a routing mechanism that selects a reasoning modality per (rule, evidence) query under an explicit cost objective, evaluated against baselines drawn from the LLM-routing literature's own evaluation standard (Section III).

A. Why routing is necessary

Building on the task definition of Section II, the system's overall decision function decomposes as $f(r_i, E) = f_{\rho(r_i,E)}(r_i, E)$, where the router $\rho(r_i, E) \rightarrow m$ selects a reasoning modality m and a modality-specific reasoner $f_m(r_i, E) \rightarrow (d, \tau)$ produces the decision and evidence trace. Formally,

$$\rho(r_i, E) \rightarrow m, \quad m \in \{\text{RULE_ENGINE},\ \text{RAG_LLM},\ \text{HUMAN_REVIEW}\}. \tag{2}$$

The three candidate modalities differ not only in cost and latency but in the correctness guarantee they can offer: $f_{\text{RULE_ENGINE}}$ is deterministic and, for a rule with complete, well-typed evidence, correct by construction relative to the extracted rule logic; $f_{\text{RAG_LLM}}$ is grounded but not deterministically guaranteed; $f_{\text{HUMAN_REVIEW}}$ defers the decision entirely. Routing is necessary because no single modality is appropriate across all seven requirement categories of Table IV: applying the rule engine to a qualitative requirement is not possible without discarding the requirement's semantic content, and applying the LLM to a fully specified numeric requirement introduces avoidable risk and cost.

B. Feature representation

The router operates on a feature vector computed per (rule, evidence) query, summarized in Table V.

C. Objective

For a candidate route m and query (r_i, E), define an expected cost

$$\begin{aligned}\text{Cost}(m) = \alpha \cdot P(\text{error} \mid m, \text{features}) \cdot \text{RiskWeight}(r_i) \\ + \beta \cdot \text{LLMTokenCost}(m) \\ + \gamma \cdot \text{Latency}(m),\end{aligned} \tag{3}$$

subject to a validation-set constraint that end-to-end compliance accuracy remains above a threshold τ_{acc} ($\tau_{\text{acc}} = 0.95$ in our experiments, enforced on held-out validation data, not on the test benchmark). The router selects the modality m minimizing $\text{Cost}(m)$ subject to this constraint. The weights α, β, and γ are tuned by grid search on a validation split (with $\alpha = 1$, the selected operating point uses $\beta = 0.1$, $\gamma = 0.001$, and a minimum evidence completeness of 0.75), and the reporting format is an accuracy–cost trade-off frontier (Section XVI) rather than a single operating point, consistent with how the LLM-routing literature reports Pareto frontiers rather than single aggregate numbers.

D. Candidate routers compared

- **R0 – Heuristic router**: the non-learned router of the implemented system (feature gates, hard override, and a heuristic choice among the routes), to which the system falls back when the adaptive router is not adopted (Fig. 2).
- **R1 – Static / rule-first**: always attempt the rule engine; fall back to the retrieval-augmented LLM only if required structured inputs are missing.
- **R2 – Confidence-threshold**: route by a single, manually tuned threshold on retrieval_confidence or rule_engine_applicability ($\theta = 0.85$ in our experiments). This is the routing strategy most systems default to in practice, and is the baseline the proposed router must beat, not merely match.
- **R3 – Fixed hybrid**: a manually authored decision table mapping rule_type to a fixed route, with no learned or per-query adaptation (corresponds to Baseline B4, Section XIV, and is denoted B4 in the results).
- **R4 – Proposed adaptive router**: a learned classifier estimating $P(\text{error} \mid m, \text{features})$ per candidate route, combined with the cost objective of Eq. (3), optionally implemented as a contextual-bandit policy over the three routes. In the evaluated configuration, the error model is a decision tree for the rule-engine route and a logistic regression for the retrieval-augmented route, each selected by validation log-loss.

Algorithm 1 presents the routing procedure. Cases with evidence completeness below a minimum threshold are escalated to human review before the cost objective is evaluated,

TABLE V
ROUTER FEATURE VECTOR.

Feature	Definition
rule_type	One-hot encoding of categories A–G (Table IV)
requires_semantic_interpretation	Extraction-time flag from the rule schema
structured_field_availability	Fraction of the rule's required evidence fields present in structured bank data
retrieval_confidence	Top-k similarity score from the vector store for the RAG path
rule_engine_applicability	1 if the rule's logic is fully deterministic and all required inputs are numeric/Boolean and present; else 0
evidence_completeness	Fraction of the rule's required evidence fields that are present and non-null
historical_router_accuracy[route]	Accuracy of each candidate route on training cases of the same rule_type
risk_level	Rule-level risk annotation (e.g., capital adequacy = high; documentation = low)

Algorithm 1 Adaptive reasoning-modality routing

Require: rule r, evidence E, trained router model M
Ensure: selected reasoning modality m^*

features $\leftarrow$ EXTRACTFEATURES(r, E) {Table V}
if features.evidence_completeness $<$ MIN_EVIDENCE_THRESHOLD **then**
  return HUMAN_REVIEW
end if
for $m \in$ {RULE_ENGINE, RAG_LLM, HUMAN_REVIEW} **do**
  $p_{\text{error}} \leftarrow M$.PREDICTERRORPROB(features, m)
  cost$[m] \leftarrow \alpha \cdot p_{\text{error}} \cdot r$.risk_weight
    $+ \beta \cdot$ ESTIMATEDLLMCOST(m, E)
    $+ \gamma \cdot$ ESTIMATEDLATENCY(m)
end for
$m_{\text{best}} \leftarrow \arg\min_m \text{cost}[m]$
if features.rule_engine_applicability $= 1$ **and** features.evidence_completeness $= 1.0$ **then**
  return RULE_ENGINE {Hard override, Sec. VIII.E}
end if
return m_{best}

since no automated modality can be trusted to produce a defensible decision from insufficient evidence. Otherwise, the expected cost of each candidate modality is computed and the minimizing modality is selected, subject to the hard override described next.

E. The hard override and its role in the correctness claim

The learned component of the router decides on the genuinely ambiguous cases; it is not what gives the deterministic guarantee for the fully specified numeric and Boolean subset of rules. The guarantee is architectural: any rule with rule_engine_applicability $= 1$ and evidence_completeness $= 1.0$ is sent by the router to the deterministic rule engine unconditionally, regardless of what the learned router would otherwise predict. This allows the system to state a correctness property that is agnostic to the accuracy of the router—one hundred percent of category A/B/C/D rules with complete evidence are answered deterministically, by construction—rather than a weaker statistical guarantee about how often the router happens to be right. This is also the specific respect in which reasoning-modality routing differs from LLM-to-LLM routing (Section III): the latter has no analogous notion of a modality that is correct by construction, because every candidate in that literature is itself a probabilistic language model.

IX. EVIDENCE-GROUNDED COMPLIANCE AND EXPLAINABILITY

Every decision produced by the system, whether generated by the rule engine, the retrieval-augmented LLM, or returned as a human-review escalation, is linked to: the source regulatory document; the specific clause; the rule identifier; the regulation version in force at the relevant date; the bank evidence fields consulted; the reasoning modality used; the resulting decision; a natural-language explanation; and, where applicable, a recommended corrective action. This evidence trace τ (Section II) is stored alongside the decision rather than reconstructed after the fact, so that an auditor can verify a decision without re-running the pipeline.

Grounding serves two purposes that are worth distinguishing. First, at generation time, constraining the retrieval-augmented LLM to cite only retrieved clauses (Section V) reduces the opportunity for the model to state a rule or threshold that does not exist in the source corpus. Second, at audit time, the stored trace allows a human reviewer to check the citation against the source document directly, independent of whether the generating model is later updated or becomes unavailable. We treat evidence-citation accuracy—whether a cited rule identifier and clause genuinely support the stated decision—and hallucination rate—the fraction of decisions citing a non-existent or incorrect clause—as measurable quantities (Section XV) rather than as an architectural property assumed to hold by design; this manuscript defines them as evaluation targets and reports no measured values for them (Section XVI).

X. END-TO-END INTEGRATION OF BOTH MODULES

Regulation Comprehension and Adaptive Compliance Checking are presented in Sections V and VII as separate modules because they serve different users and produce different outputs, but they are not two disconnected applications: they operate over the same regulatory knowledge base, and the structured rule representation produced once during comprehension is read, not regenerated, by the compliance module. The dependency runs in one direction only, illustrated below:

RBI regulation $\rightarrow$ comprehension $\rightarrow$ structured rules $\rightarrow$ searchable regulatory knowledge $\rightarrow$ compliance evaluation.

The same extracted regulatory representation therefore serves two purposes: a human-facing purpose, in which a compliance officer or auditor queries the knowledge base through natural language and receives a cited, simplified answer (Section V); and a machine-facing purpose, in which the same typed rule objects are consumed by the router and reasoning engines of Section VIII to evaluate bank evidence.

This sharing has two practical consequences that would not hold if the two modules maintained independent extraction pipelines. First, a correction to a rule's extracted representation—for example, following human verification (Section V) or an amendment update—propagates automatically to both the question-answering interface and the compliance decision engine, rather than requiring the correction to be made twice and risking the two representations drifting apart. Second, the version and effective-date metadata recorded once during comprehension (Section V) is the same metadata that resolves cross-version compliance test cases (Section XII), so that a query about "what does the current regulation require" and a compliance check against "the regulation in force on a given date" are answered from a single consistent source rather than two representations that could disagree.

We are explicit that this integration, by itself, is not presented as the paper's primary novel contribution: connecting a comprehension module to a downstream reasoning module is a natural systems-engineering choice rather than a new technique. It is presented as a system-level contribution that makes the primary contribution—adaptive reasoning-modality routing (Section VIII)—practically deployable within a single coherent regulatory intelligence workflow, rather than as a standalone research claim.

XI. System Architecture and Implementation

This section describes the implementation supporting the architecture of Section IV, distinguishing components that are implemented from components that remain proposed or future work.

A. Relational and vector storage

Structured rules, rule versions, clauses, source documents, applicability conditions, and compliance results are normalized into a relational schema in PostgreSQL, following the fields defined in Section VI; the prototype used for the experiments and shown in Fig. 1 runs the same schema on an embedded SQLite store with FTS5 keyword search, which is sufficient at the corpus scale of Section XVI. Clause and definition embeddings for semantic retrieval are indexed using pgvector, co-located with the relational store rather than in a standalone vector database, given a corpus scale (thousands, not billions, of vectors) consistent with the benchmark scale discussed in Section XII.

B. Rule engine

The deterministic reasoning modality evaluates the logic block of each rule object directly using a JSON-Logic-style rule engine, bypassing both a Java-virtual-machine-based rule engine and a custom domain-specific language which would itself have to be validated.

C. LLM with structured output and function calling

Two uses of a large language model are implemented: schema-constrained extraction (Section V), in which the model is constrained to emit valid instances of the schema of Section VI, and retrieval-augmented reasoning (Sections V and VIII), in which the model reasons over retrieved clauses under a citation-constrained prompting strategy. Model identity and version are pinned and recorded (Section XII) rather than left unspecified, since extraction and routing results are expected to shift if the underlying model is updated.

D. Embeddings and retrieval

Clause and definition text is embedded with a fixed, recorded embedding model (Section XII) and retrieved by vector similarity, feeding both the question-answering interface of Section V and the retrieval-augmented reasoning path of Section VIII.

E. Regulatory document and evidence processing

Document comprehension (Section V) and bank-evidence comprehension (Section VII) make use of format-specific parsers (layout-aware PDF/HTML extractors for regulatory documents; CSV/JSON/PDF-text parsers mapped to a canonical schema for bank evidence), rather than a single generic parser, due to the significant differences in structure and reliability between the two input formats.

F. Offline versus query-time processing

Regulatory-side processing (document ingestion, clause segmentation, rule extraction, and knowledge-base population) is either offline or asynchronous, since it is driven by newly ingested or amended RBI documents. Compliance-side processing (evidence ingestion, routing, reasoning, and decision assignment) is done at query time, against a regulatory knowledge base populated offline. This way, there is no query-time dependency on latency-heavy document processing.

G. Implemented versus proposed components

The relational and vector storage layers, the deterministic rule engine, schema-constrained extraction, retrieval-augmented question answering and reasoning, and the router's feature extraction (Table V) are implemented components of the architecture as described. The learned cost-minimizing router policy of Section VIII (as distinct from its feature representation and hard override, both of which are architectural rather than learned) and the connected-counterparty graph reasoning of Section VI (category G) are, respectively, evaluated on the synthetic benchmark (Section XVI) and evaluated only on a small synthetic subset of cases (Section XVI); we do not present either as validated on real bank data.

XII. Experimental Methodology

Consistent with treating both modules as independently measured contributions (Section V), this section presents two methodologies side by side: Section XII.A–F covers Adaptive Compliance Checking, and Section XIII covers Regulation Comprehension. Both follow the same construction discipline—systematically derived from source material rather than authored independently of it, with disclosed provenance—so that the two sets of results, once obtained, are comparable in rigor even though they measure different capabilities.

A. Benchmark construction (compliance checking)

Given the confidentiality of banks' operational data, which is unavailable for academic use (Section XIX), a synthetic benchmark is systematically constructed from the extracted rule corpus (Section VI) rather than being generated independently of it, ensuring that each test case can be traced back to a particular rule_id and its source clause, following recent calls in the financial-LLM-evaluation literature for benchmarks going beyond static question-answer pairs to include boundary, adversarial, and escalation scenarios [23]–[25].

B. Case categories (compliance checking)

Table VI summarizes the nine case-difficulty categories used to construct the benchmark.

C. Case schema and provenance (compliance checking)

Each case is saved as a record with fields case_id, entity_info, rule_id, input_data, relevant_documents, expected_decision, expected_reason, difficulty_level, case_type, regulation_version, and generation_method. We retain the generation_method field (one of template, LLM-perturbation, or human-authored) to reflect the provenance of each record in the benchmark as a whole (recent work explicitly asks synthetic-benchmark papers to disclose this rather than presenting synthetic data as though it were similar to naturally occurring data [23], [25]).

D. Scale (compliance checking)

The benchmark is deliberately bounded rather than attempting exhaustive coverage of RBI regulation: it contains 1,393 cases drawn from 215 rules, with the composition by case type and by requirement category shown in Fig. 4, so that per-case construction quality and, where feasible, human review remain tractable for a single-team research project.

E. Training, validation, and test separation

The compliance benchmark is split by rule into training, validation, and test partitions (840, 259, and 294 cases, respectively; Fig. 4) in such a way that the learned router policy (Section VIII) and its cost weights α, β, γ are tuned only on the training and validation partitions, with the accuracy constraint τ_{acc} (Section VIII) enforced on the validation partition, and all reported figures (Section XVI) computed on the held-out test partition. The comprehension benchmark (Section XIII) is split identically in spirit—by source document, so that no clause from a test-partition document is ever seen in extraction-prompt few-shot examples tuned on the training partition.

F. Implementation details for reproducibility

The following details were recorded with the evaluation: the relational schema is implemented in PostgreSQL (SQLite in the evaluated prototype, Section XI), normalized into rules, rule_versions, clauses, documents, applicability_conditions, and compliance_results tables (Section XI); the vector store is pgvector, co-located with the relational store; the rule engine executes the logic block of each rule object directly via a JSON-Logic-style engine; the large language model is Gemini 2.5 Flash (temperature 0, JSON output) and the embedding model is BGE-small-en-v1.5 (Fig. 1); and the manually verified rule-extraction sample belongs to the comprehension evaluation (Section XIII), whose results are not reported here. So that the benchmark runs are reproducible without API access, the runs reported in Section XVI replace the LLM call on the retrieval-augmented path and on the LLM-only baseline with a deterministic surrogate judge that follows the same citation-constrained instructions (the requirement's threshold and direction, the reported figure, negation, hedging, contradiction, and clause relevance); LLM cost and latency in these runs are modelled from Gemini 2.5 Flash list prices and an assumed 1.8 s per call, and a human review is costed at an assumed 10 minutes at USD 9 per hour (USD 1.50 per case).

XIII. Experimental Methodology for Regulation Comprehension

Module 1 is evaluated on a separate held-out benchmark (rather than the compliance-checking benchmark of Section XII) because it tests a different competency: faithful extraction, retrieval, and explanation of what a regulation says, regardless of downstream compliance implications.

A. Benchmark construction (regulation comprehension)

The comprehension benchmark is constructed from the same ingested RBI Master Directions and circulars that populate the shared regulatory knowledge base (Section X), so that every comprehension test case, like every compliance test case, traces back to a specific source clause. Three parallel sub-benchmarks are constructed: an *extraction* sub-benchmark of clause-level ground truth for segmentation, definition linking, obligation typing, and category labels; a *question-answering* sub-benchmark of natural-language questions, each with a gold answer, gold supporting clause(s), and gold regulation version; and a *simplification* sub-benchmark of clause/simplification pairs with expert-authored reference simplifications. This mirrors, at a construction level, how expert-annotated legal datasets such as CUAD [13] and numerical financial-QA datasets such as FinQA [14] pair source documents with span- or answer-level ground truth rather than treating the source text and the ground truth as independently collected.

TABLE VI
BENCHMARK CASE CATEGORIES AND CONSTRUCTION METHOD (ADAPTIVE COMPLIANCE CHECKING).

Case type	Construction method
Compliant / Non-compliant	Evidence values sampled clearly on the satisfying / violating side of the rule's threshold or condition.
Boundary	Evidence value exactly equal to the threshold; forces an explicit, documented convention for $\leq$ versus $<$ per rule.
Near-boundary	Evidence within $\pm$1–5% of threshold, testing sensitivity and robustness rather than gross correctness.
Ambiguous	Evidence for qualitative (category E) rules that does not unambiguously satisfy or violate the stated language; expected_decision = AMBIGUOUS (escalation to human review is also accepted), not a forced binary label.
Missing-data	Required evidence fields deliberately withheld; expected_decision = INSUFFICIENT_DATA, testing whether a system fabricates a decision under missing evidence.
Adversarial	Evidence phrased to appear compliant in qualitative language while numerically failing (or vice versa), testing whether surface phrasing can override a deterministic check that should dominate.
Contradictory	Two evidence sources disagree; expected_decision = AMBIGUOUS or REQUIRES_HUMAN_REVIEW with both sources cited.
Cross-version	Identical fact pattern evaluated against two rule_versions (pre-/post-amendment), with different expected outcomes, testing versioning and effective-date logic.

B. Extraction sub-benchmark

The ground truth for clause segmentation, definition extraction, obligation-type labels (obligation/prohibition/permission/definition/explanatory, Section V.F), and category labels (Section V.G) is obtained by manual annotation of a disclosed sample of the ingested documents, using the same general approach as for rule-extraction verification in Section XII.F: the sample size and source documents should be specified explicitly rather than inferred from the data, and the annotation should be performed independently, and preferably before running the extraction pipeline over the same clauses to be annotated, to avoid the possibility of the pipeline's outputs contaminating its evaluation.

C. Question-answering sub-benchmark

Natural-language questions are created in three forms, each marked by its `generation_method` field analogous to the compliance benchmark (Section XII.C): template questions mechanically generated from a rule's applicability and antecedent fields ("what is the exposure limit for a single counterparty"); paraphrase questions that rephrase a template question but retain the same target clause, testing retrieval robustness to lexical variation; and multi-clause questions that require combining two or more clauses (e.g., a general rule and a stated exception), testing whether retrieval returns a sufficient evidence set rather than only the single best-matching clause. Each question records a gold answer, the gold supporting clause identifier(s), and, for cross-version questions, the regulation version the question is posed against.

D. Simplification sub-benchmark

A subset of all extracted clauses, stratified across the seven requirement categories listed in Table IV (so that not only the simplest, quantitative clauses are used for evaluation), is paired with a reference simplification. The reference simplifications in this sub-benchmark are written by a different party than the one that tunes the simplification prompt, to avoid accidental encoding of the reference simplification strategy into the prompt itself.

E. Intended scale (regulation comprehension)

The extraction sub-benchmark targets full-coverage annotation of the same Master Directions used for the compliance benchmark (Section XII.D); the question-answering sub-benchmark combines template, paraphrase, and multi-clause questions; and the simplification sub-benchmark uses clause/reference pairs stratified across categories A–G. The exact numbers of questions and pairs, and the ratio across question types, are reported together with the comprehension results, which are outside the scope of this manuscript.

F. Evaluation protocol

Extraction metrics (Section XV-B, item A) are calculated by comparing pipeline output against the extraction sub-benchmark's manual annotations. Retrieval and question-answering metrics (Section XV-B, item B) are calculated against the question-answering sub-benchmark, retrieval being evaluated independently of generation (whether the correct clause appears in the top-k retrieved set) before end-to-end answer quality is evaluated, so that a retrieval failure and a generation failure are never conflated into a single opaque number. Simplification metrics (Section XV-B, item C) are calculated against the simplification sub-benchmark. As with the compliance benchmark, no model or prompt tuning is permitted against the held-out test partition of any of the three sub-benchmarks.

XIV. BASELINES

A. Baselines for Adaptive Compliance Checking

Table VII summarizes the four baselines and the proposed system, each isolating a different design choice.

B. Baselines for Regulation Comprehension

Module 1 is compared against three baselines capturing increasingly sophisticated comprehension architectures, summarized in Table VIII, to tease out the contribution of RBI-specific structuring (Section V) from that of retrieval-augmented generation at large [9].

TABLE VII
BASELINES COMPARED AGAINST THE PROPOSED SYSTEM (ADAPTIVE COMPLIANCE CHECKING).

Baseline	Description	Known limitation
B1 – Rule-only	Structured rules executed by the deterministic engine only.	No coverage of qualitative (category E) requirements; brittle to missing or unstructured evidence.
B2 – LLM-only	Raw bank data and regulatory text passed directly to the LLM with no retrieval or rule-engine grounding.	No grounding guarantee; expected to be most exposed to hallucination on adversarial and missing-data cases.
B3 – RAG+LLM	Vector retrieval over regulatory clauses feeding LLM reasoning, with citation.	Adds grounding over B2 but no deterministic guarantee on the numeric subset; routes everything through the LLM regardless of determinism.
B4 – Fixed hybrid	Rule engine and RAG+LLM present, but routed by a hand-authored fixed rule_type $\rightarrow$ route table, with no learning or per-query adaptation.	Same reasoning engines as the proposed system; isolates the value of adaptive routing specifically.
Proposed	Rule engine + RAG+LLM + human-review escalation, governed by the adaptive router (Section VIII) with the hard-override grounding guarantee.	Bounded by extraction quality (RQ3) and by benchmark realism (Section XIX).

TABLE VIII
BASELINES COMPARED AGAINST THE PROPOSED SYSTEM (REGULATION COMPREHENSION).

Baseline	Description	Known limitation
C1 – Keyword search	Lexical (BM25-style) search over raw extracted document text, with no clause structuring, no embeddings, and no generation; the user reads the retrieved passage directly.	No answer synthesis or simplification; fails on paraphrased queries with no lexical overlap with the source clause.
C2 – LLM-only (no retrieval)	The question is answered directly by the LLM from its parametric knowledge, with the raw document supplied in-context only up to a fixed context budget, without clause-level retrieval or citation constraints.	No grounding guarantee; expected to be most exposed to hallucination on long documents that exceed the context budget and on multi-clause questions.
C3 – Generic RAG	Standard retrieval-augmented generation [9] over the document with generic chunking (fixed-length windows), without RBI-specific clause segmentation, definition linking, category classification, or version/effective-date metadata (Section V).	Retrieves topically relevant but structurally misaligned chunks (e.g., splitting a clause mid-sentence); no versioning or cross-reference resolution.
Proposed	Layout-aware ingestion, clause segmentation, definition linking, and category classification (Section V) feeding structured retrieval and citation-constrained generation, with version-aware resolution.	Bounded by extraction quality (RQ3) and by the disclosed scale of the comprehension benchmark (Section XIII.E).

XV. EVALUATION METRICS

A. Metrics for Adaptive Compliance Checking

Metrics are grouped by research question rather than reported as an undifferentiated pool.

RQ1/RQ4 (compliance accuracy and grounding): accuracy, precision, recall, F1, false-positive rate, false-negative rate, evidence-citation accuracy (whether the cited rule_id and clause actually support the decision), and hallucination rate (the fraction of decisions citing a non-existent or incorrect clause), each reported overall and stratified by case_type (Table VI). Two accuracy measures are reported in Section XVI: automated accuracy, in which a case counts as correct only if the produced decision is in the case's set of acceptable decisions, so that an escalation counts as undecided except on ambiguous and contradictory cases, for which human review is an acceptable outcome (Table VI); and end-to-end accuracy, the counterpart measure that includes the human-review path and assumes that the reviewer resolves every escalated case correctly. For a decision set with true positives TP, false positives FP, true negatives TN, and false negatives FN,

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \tag{4}$$

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

RQ2 (routing efficiency): the accuracy–cost–latency Pareto frontier across routers R0–R4 (Section VIII); routing accuracy against a hand-labeled "correct route" reference set; and the fraction of LLM calls avoided relative to B2/B3.

RQ3 (extraction quality): entity, operator, threshold, and unit extraction precision, recall, and F1, and rule executability rate, all broken down by requirement category A–G (Table IV).

B. Metrics for Regulation Comprehension

Metrics for Module 1 are likewise grouped by the sub-benchmark of Section XIII they are computed against, rather than pooled into a single comprehension score.

A. Extraction quality: clause-segmentation boundary accuracy (fraction of predicted clause boundaries matching the manual annotation exactly); definition-linking precision and recall (whether a defined term is correctly linked to its governing definition); obligation-type classification accuracy and category-classification F1 (both against the manual labels of Section XIII.B), reported per requirement category A–G as in Table IV, since we expect segmentation and typing to be markedly easier for quantitative (category A) clauses than for qualitative (category E) or relational (category F) ones.

B. Retrieval and question-answering quality: retrieval is evaluated independently of generation using top-k recall

(whether the gold supporting clause appears among the top-k retrieved clauses) and mean reciprocal rank (MRR),

$$\text{MRR} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\text{rank}_q}, \tag{6}$$

where rank_q is the rank of the first correctly retrieved clause for question q. End-to-end answer quality is then evaluated using the retrieval-augmented-generation decomposition of RAGAs [11]: *context precision* and *context recall* (retrieval-side), and *faithfulness* (the fraction of claims in the generated answer that are supported by the retrieved clauses) and *answer relevance* (generation-side). Citation accuracy (whether the cited clause identifier genuinely supports the stated answer, evaluated directly against the gold supporting clause rather than only via faithfulness) and hallucination rate (the fraction of answers citing a non-existent or incorrect clause identifier) are reported using the same definitions as the corresponding compliance-side metrics (Section XV.A), following the broader hallucination-evaluation framing of Ji et al. [12], so that grounding quality is comparable, not merely analogous, across the two modules.

C. Simplification quality: SARI [29], which scores a simplification's inserted, deleted, and kept n-grams against one or more reference simplifications and is, to our knowledge, the most widely used automatic metric for this purpose; the Flesch–Kincaid Grade Level readability score, to check that the simplification is in fact easier to read rather than merely shorter; and a meaning-preservation check comparing the simplification's stated conditions and exceptions against the original clause's, since neither SARI nor a readability score on its own would catch a simplification that reads well but silently drops an exception (Section V.H).

D. Cross-module consistency: the fraction of clauses for which the category label and version metadata used by the question-answering interface (this section) agree exactly with the category label and version metadata used by the compliance decision engine (Section XV.A) for the same underlying rule_id, which should be 1.0 by architectural construction (Section X) since both read the same stored object; a measured value below 1.0 would indicate a synchronization defect between the two modules' read paths rather than a modeling error, and is reported specifically to make that distinction checkable.

XVI. Results and Discussion

This section reports results for the Adaptive Compliance Checking module (Module 2) on the synthetic benchmark of Section XII. Unless stated otherwise, all results are computed on the held-out test partition of 294 cases, whose rules were never seen in training or tuning. Numbers printed in the figures are quoted exactly; numbers introduced with "approximately" are read from marker positions in the figures. Of the Regulation Comprehension module (Module 1), only the composition of the extracted corpus is reported here; no result of the comprehension benchmark of Section XIII (retrieval, question answering, simplification, per-category extraction accuracy) is reported in this manuscript.

In the figures, B1–B3 are the baselines of Table VII, and B4 is the fixed hybrid, which is router R3 of Section VIII.D. R0 denotes the non-learned heuristic router that the system falls back to when the adaptive router is not adopted (Fig. 2), R1 the static rule-first router, R2 the confidence-threshold router, and R4 the proposed adaptive router. R4 is shown at several operating points obtained with different cost weights (labelled "R4 at R0's cost" and "R4 fully automated" in the figures), and the headline R4 operating point is the one labelled "R4 (proposed)" in Fig. 5 and "R4 (τ)" in Fig. 6. *Automated accuracy* counts a decision as correct when it is in the case's set of acceptable decisions, so an escalation counts as undecided except on ambiguous and contradictory cases, where human review is acceptable; *end-to-end accuracy* is the counterpart measure that includes the human-review path and assumes the reviewer resolves every escalated case correctly.

A. Corpus and benchmark composition

Fig. 3 shows the distribution of the 14,838 obligation clauses extracted from the corpus (264 RBI directions across 14 institution categories, 21,632 clauses in total) over the seven requirement categories: qualitative clauses (E) dominate with 11,741 clauses (79.1%), followed by temporal (D, 1,341; 9.0%), relational (F, 815; 5.5%), quantitative (A, 438; 3.0%), Boolean/procedural (B, 399; 2.7%), multi-entity (G, 65; 0.4%), and conditional (C, 39; 0.3%). Most of the regulatory text therefore falls outside the reach of a purely deterministic engine, while the numeric and Boolean clauses that can be handled deterministically are the minority.

The compliance benchmark (Fig. 4) contains 1,393 cases from 215 rules, split by rule into 840 training, 259 validation, and 294 test cases. By case type it contains 207 non-compliant, 203 compliant, 190 missing-data, 189 adversarial, 173 contradictory, 150 ambiguous, 146 near-boundary, 89 boundary, and 46 cross-version cases. By requirement category it contains 487 (A), 216 (B), 80 (C), 192 (D), 288 (E), 96 (F), and 34 (G) cases. The benchmark's category mix differs deliberately from the corpus mix: category A makes up 35.0% of the cases but 3.0% of the clauses, and category E makes up 20.7% of the cases but 79.1% of the clauses, so that the deterministic categories are exercised in depth. The test partition contains 109, 12, 15, 48, 84, 18, and 8 cases of categories A–G, respectively (Fig. 8); the smaller categories therefore support only indicative conclusions.

B. Overall accuracy and routing comparison (RQ1, RQ2)

Fig. 5 compares the baselines and routers on the test partition. The single-mechanism baselines reach 66.0% (B1, rule-only), 56.8% (B2, LLM-only), and 72.5% (B3, RAG+LLM). Grounding the LLM in retrieved clauses (B2 to B3) is associated with an improvement of 15.7 points, and adding the rule engine under a fixed rule-type-to-route table (B3 to B4) with a further 12.9 points, to 85.4%. The proposed adaptive router (R4) reaches 92.2% automated and 96.9% end-to-end

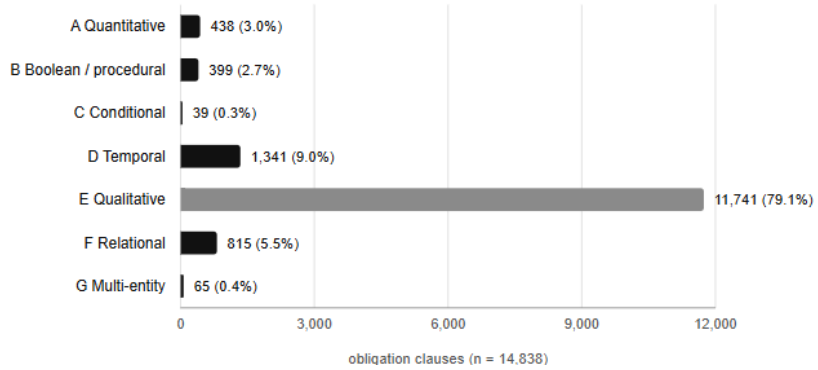


Fig. 3. Distribution of the extracted obligation clauses over requirement categories A–G ($n = 14{,}838$).

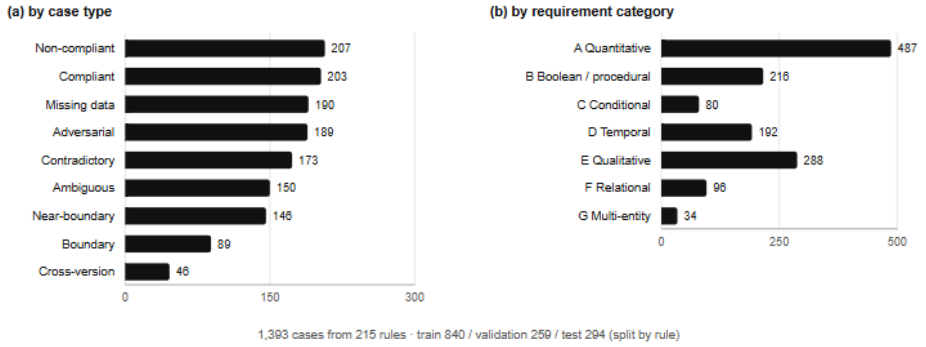


Fig. 4. Composition of the synthetic compliance benchmark (1,393 cases from 215 rules; train 840 / validation 259 / test 294, split by rule): (a) by case type and (b) by requirement category.

accuracy, 6.8 points above B4 in automated accuracy, 8.9 points above the heuristic router R0 (83.3%), and 5.8 points above the static rule-first router R1 (86.4%). The confidence-threshold router R2 reaches only 71.8% automated accuracy but 95.6% end-to-end accuracy, a gap of 23.8 points that indicates a heavy reliance on escalation; the corresponding gaps are 9.6 points for R0 and 4.7 points for R4. R4 is 2.0 points below the oracle's automated accuracy of 94.2%. Two other R4 operating points are shown: a fully automated one at 87.8%, above R1 (86.4%) and B4 (85.4%) without any escalation, and one at R0's cost that reaches 99.0% end-to-end accuracy against 92.9% for R0. On a paired cluster bootstrap over rules (2,000 resamples), the 95% confidence interval of R4's automated-accuracy gain is [4.3, 9.4] points over B4 and [5.3, 12.7] points over R0 (exact McNemar $p < 10^{-5}$ in both cases), whereas the 1.4-point gain of the fully automated operating point over R1 is not significant ($p = 0.22$).

C. Accuracy–cost trade-off (RQ2)

Fig. 6 plots end-to-end accuracy against cost per case on a logarithmic axis, together with 72 tuned R4 configurations. The cost-free rule-only baseline B1 (66.0%) and R4's fully automated operating point (87.8%) bound the cheap end of the frontier, at a cost of approximately 2×10^{-4} US dollars per case for the latter, essentially the same as R1 and B4. Operating points that use human escalation lie roughly three orders of magnitude to the right of the automated ones, under the cost model of the evaluation harness. In that region the R4 operating point at 96.9% costs less per case than R0 (92.9%) and R2 (95.6%) while being more accurate than both, and the R4 operating point at R0's cost reaches 99.0%. B2 and B3 lie well below the frontier. Hence every baseline and alternative router other than the zero-cost B1 is dominated by at least one R4 operating point in this plot. Latency and the fraction of LLM calls avoided, which are part of the metrics of Section XV, are not reported in the figures and are therefore not claimed here.

D. Robustness by case type (RQ5)

Fig. 7 shows automated accuracy by case type for R4, R0, B4, and B3. R4 reaches 100% on the compliant (40 cases), non-compliant (46), boundary (20), near-boundary (34), missing-data (39), contradictory (36), and cross-version (10) cases. Two case types remain difficult: ambiguous cases, where R4 reaches approximately 80% (through escalation to human review, which is acceptable for this case type) against approximately 67% for R0, approximately 40% for B3, and 20% for B4; and adversarial cases, where R4, R0, and B4 all lie at approximately 56% and B3 at approximately 46%. That three differently routed systems coincide on the adversarial cases suggests that routing is not the main source of the remaining errors on this case type, although the figure alone

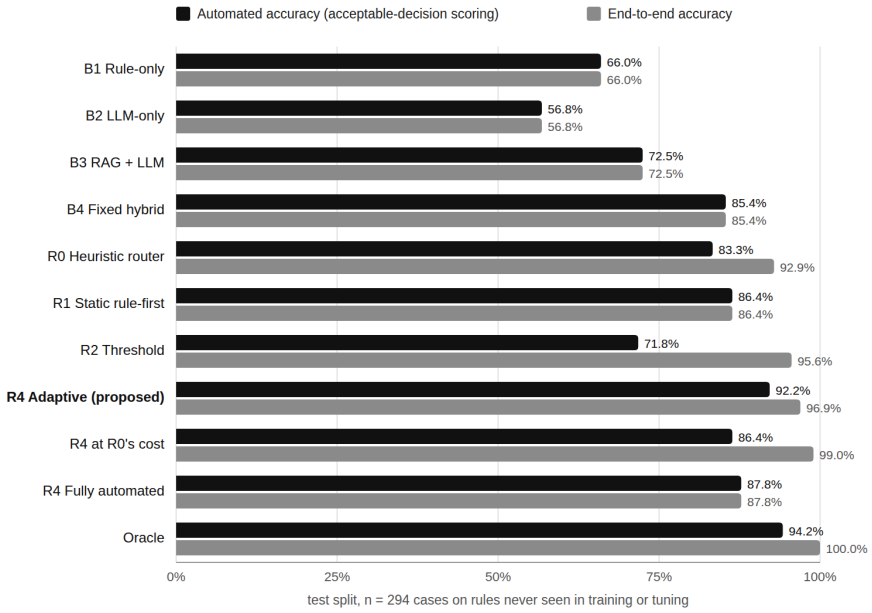


Fig. 5. Automated and end-to-end accuracy of the baselines (B1–B4), routers (R0–R2, R4), two further R4 operating points, and the oracle on the test partition ($n = 294$ cases on rules never seen in training or tuning). Automated accuracy uses the acceptable-decision scoring of Section XV.

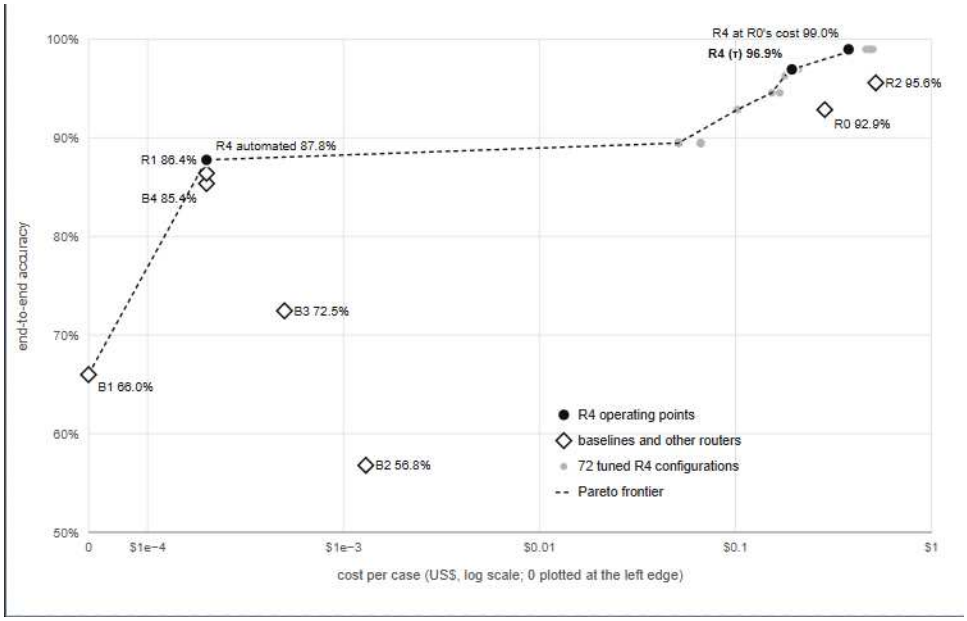


Fig. 6. End-to-end accuracy versus cost per case (US$, logarithmic scale; zero cost plotted at the left edge) for R4 operating points, the baselines and other routers, and 72 tuned R4 configurations; the dashed line is the Pareto frontier.

does not identify their cause. Differences between systems are largest where a fixed route is a poor fit, for example ambiguous cases for B4 and cross-version cases for B3 (approximately 50%), but the cross-version and boundary subsets contain only 10 and 20 test cases.

E. Accuracy and routing by requirement category

Fig. 8 shows automated accuracy by requirement category. R4 reaches 100% on the Boolean/procedural (12 cases), conditional (15), temporal (48), and multi-entity (8) categories, approximately 97% on the quantitative category (109), approximately 80% on the qualitative category (84), and approximately 83% on the relational category (18). On the qualitative and relational categories R4 is above R0 (approximately 62% and 56%), B4 (approximately 71% and 78%), and B3 (approximately 71% and 78%). B3, which has no rule engine, reaches approximately 66% on the quantitative category and approximately 12% on the multi-entity category,

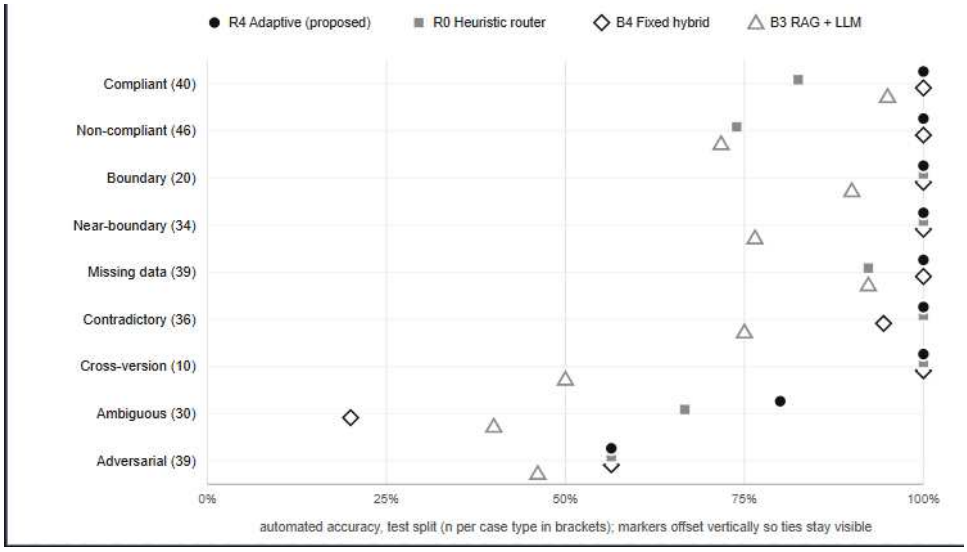


Fig. 7. Automated accuracy by case type on the test partition for R4, R0, B4, and B3 (number of cases per type in brackets; markers are offset vertically so that ties stay visible).

which illustrates the cost of sending fully specified numeric requirements through a language model.

Fig. 9 shows how R4 routes the test cases. Conditional (C) and multi-entity (G) cases are sent entirely to the rule engine or graph check, and temporal (D) cases predominantly so; the quantitative category (A) is also routed mostly to the rule engine, with the remainder split between human review and cases for which the system declined because evidence was missing. Qualitative (E) and relational (F) cases are routed to the RAG+LLM path, human review, or declined, and none to the rule engine, while the small Boolean/procedural category (B) is split across all four outcomes. This matches the preferred reasoning modalities of Table IV.

F. Ablation (RQ1)

Fig. 10 reports the compliance-checking ablations. The full system reaches 92.2% automated and 96.9% end-to-end accuracy. Removing adaptive routing (which reduces to B4) lowers automated accuracy by 6.8 points; removing the rule engine (reducing to B3) by 19.7 points; and removing evidence grounding (reducing to B2) by 35.4 points. Because these three configurations are the baselines B4, B3, and B2, each removes more than the named component (B2, for example, also lacks the rule engine and routing), so these differences bound rather than isolate that component's contribution. Replacing the learned error model with historical route accuracy only lowers automated accuracy by 7.8 points and end-to-end accuracy by 2.3 points, and removing the evidence-completeness gate lowers them by 0.4 and 0.3 points. Removing the hard override leaves both measures unchanged on this test partition. This is consistent with the role given to the override in Section VIII.E: it guarantees that fully specified rules with complete evidence reach the deterministic engine, and it is not expected to raise accuracy when the learned router already makes the same choice on these cases.

G. Error structure

Fig. 11 compares the decisions produced by R4 and R0 against the expected decisions. R4 reproduces all 91 compliant and all 39 insufficient-data cases, labels 80 of the 97 non-compliant cases correctly, and escalates 14 and labels 3 as ambiguous among the remaining 17. For the 30 ambiguous cases, R4 escalates 24 to human review and does not produce the ambiguous label on the diagonal, whereas R0 does so for 5. Under the acceptable-decision scoring of Section XV, these 24 escalations count as correct automated outcomes; counting them as undecided would place R4's automated accuracy at 84.0%. For the 37 cases whose expected decision is human review, R4 escalates 18 and labels 19 as ambiguous, whereas R0 escalates 31 and labels 6 as ambiguous. R0 escalates 7 compliant cases, labels 11 non-compliant cases as ambiguous, and escalates 18 of them. R4's gains in the earlier figures thus come largely from fewer unnecessary escalations and ambiguous labels on decidable cases, while it is less faithful than R0 in reproducing the expected "human review" label itself.

H. Discussion

On this benchmark, the results are consistent with the premise of the paper that no single reasoning mechanism suffices (RQ1, RQ2): the three single-mechanism baselines are the weakest systems, a fixed hybrid recovers a large part of the gap, and the adaptive router adds a further improvement together with a better accuracy–cost trade-off, with the largest gains on qualitative, relational, and (through escalation) ambiguous cases. The ablations show the largest accuracy losses for the configurations without evidence grounding and without the rule engine, and a substantial loss to removing either adaptive routing or the learned error model, whereas the evidence-completeness gate and the hard override have

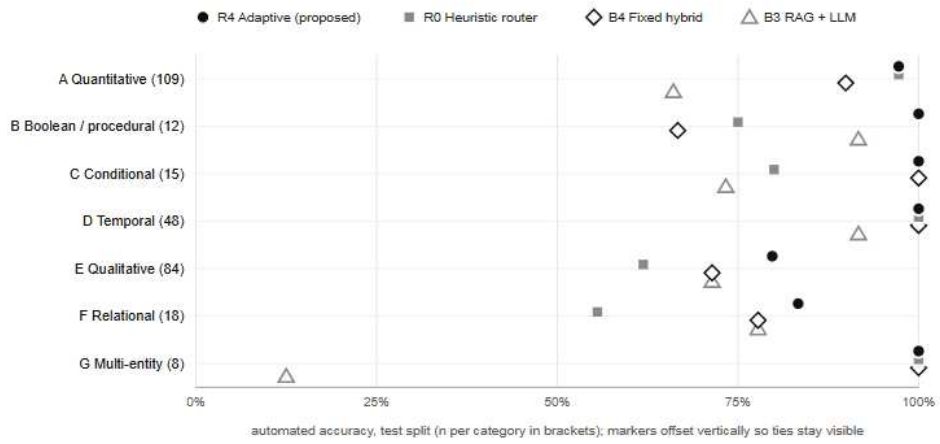


Fig. 8. Automated accuracy by requirement category on the test partition for R4, R0, B4, and B3 (number of cases per category in brackets; markers are offset vertically so that ties stay visible).

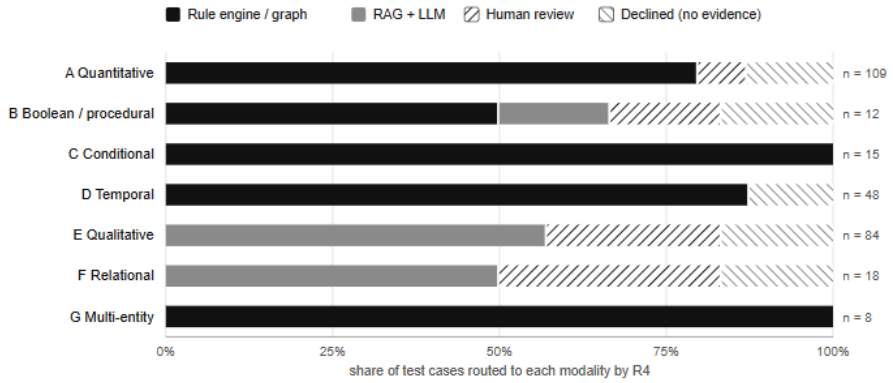


Fig. 9. Share of test cases routed by R4 to each modality (rule engine / graph, RAG+LLM, human review, or declined for lack of evidence), by requirement category.

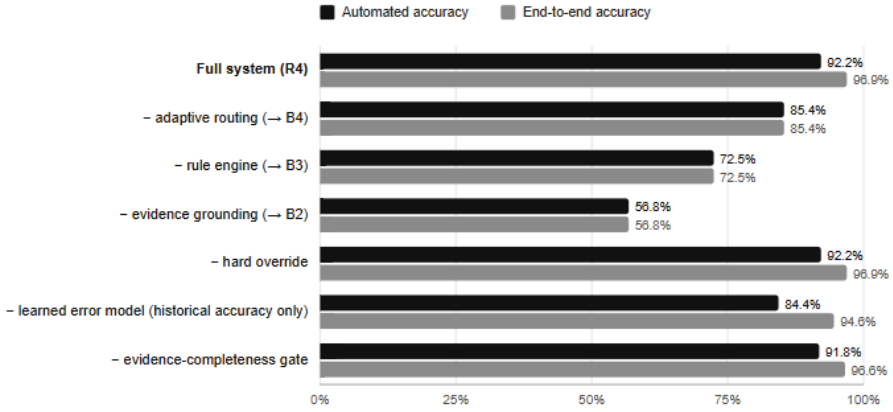


Fig. 10. Ablation of the full system (R4): automated and end-to-end accuracy when adaptive routing, the rule engine, evidence grounding, the hard override, the learned error model, or the evidence-completeness gate is removed.

little or no effect on accuracy in this test partition. The architectural properties of Sections VIII and IX—deterministic handling of fully specified rules, escalation under insufficient evidence, and a stored evidence trace for every decision—remain properties of the design; in particular, the correctness of the deterministic path is relative to the extracted rule logic and is therefore bounded by extraction quality, which is not reported per category in this manuscript.

The evidence has clear limits. All results are on a synthetic, rule-grounded benchmark and not on real bank data, and

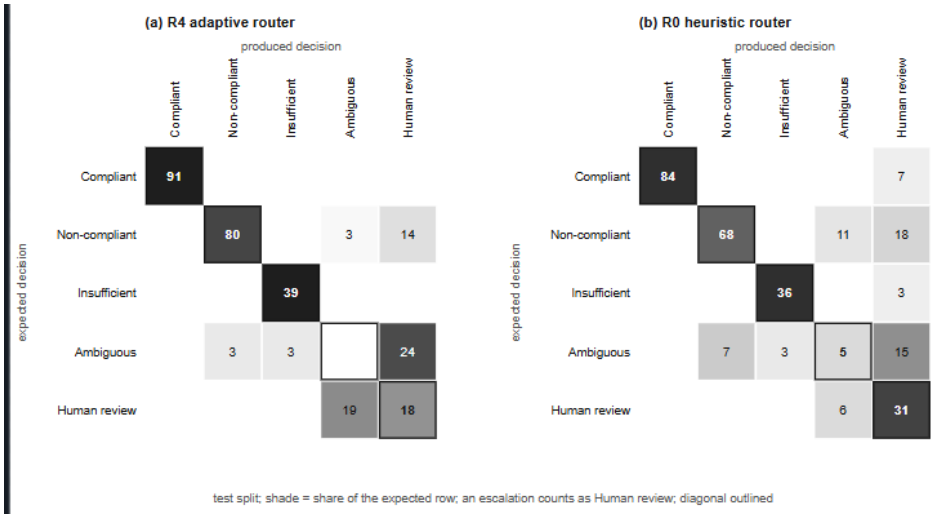


Fig. 11. Confusion matrices on the test partition for (a) the R4 adaptive router and (b) the R0 heuristic router; rows are expected decisions, columns are produced decisions, shade is the share of the expected row, an escalation counts as human review, and the diagonal is outlined.

the category-G results, which concern graph-based exposure aggregation, rest on 8 test cases built from author-generated data. The decisions of the LLM paths come from the surrogate judge of Section XII.F, so the B2 and B3 figures characterise the routing framework rather than the accuracy of a particular LLM. The test partition has 294 cases, several subgroups are small, results come from a single rule-level split, and confidence intervals and significance tests are reported only for the main router comparisons, so other differences of a few points should not be over-interpreted. The R4 configurations were tuned over a grid of 72 configurations (Fig. 6), and the cost model behind the cost axis is that of the evaluation harness. Adversarial and ambiguous cases remain the weakest, citation accuracy, hallucination rate, latency, and per-category extraction accuracy are not reported, and RQ3, RQ4, RQ6, and RQ7 are therefore not answered by this manuscript.

XVII. Ablation Study

A. Ablations for Adaptive Compliance Checking

The results of the compliance-checking ablations are reported in Fig. 10 and discussed in Section XVI. The ablations isolate, respectively: the marginal contribution of adaptive routing over a fixed hybrid decision table (reduces to B4); the contribution of the deterministic rule engine over retrieval-augmented reasoning alone (reduces to B3); the contribution of evidence grounding over ungrounded LLM reasoning (reduces to B2); the hard override of Section VIII.E; the learned error model, replaced by historical route accuracy only; and the evidence-completeness gate of Algorithm 1. The planned ablation of the deterministic validation step in rule extraction (Section V) is not reported. An ablation isolating a "no graph module" condition is omitted because category G has too few test cases (8) to support it, and a "no confidence mechanism" condition is omitted because the confidence-threshold comparison is already captured by baseline router R2 (Section VIII.D).

B. Ablations for Regulation Comprehension

A parallel set of ablations is planned to isolate the contribution of the comprehension-specific design choices of Section V, but has not been run and no result is reported: the contribution of layout-aware ingestion (Section V.B) over flat-text extraction; the contribution of RBI-specific clause segmentation over generic fixed-length chunking (reduces retrieval to baseline C3, Table VIII); the contribution of definition linking (Section V.E) specifically to simplification quality, since an unlinked defined term is a common source of an under-specified simplification; and the contribution of the citation constraint on generation (Section V.N) to faithfulness, with retrieval metrics not applicable in that configuration since removing the citation constraint does not by itself change what is retrieved.

XVIII. Novelty and Contribution Analysis

Each candidate contribution is assessed against the related work of Section III individually, rather than asserting novelty uniformly across the system, consistent with the principle that implementation technology—a specific database or a specific rule-engine library—is not itself a research contribution. Table IX summarizes the assessment.

The paper's contribution claim rests principally on Section VIII (the router, evaluated in Section XVI) and secondarily on Sections V and XII (extraction and benchmark; Section XVI reports results on the benchmark but not per-category extraction accuracy), and not on the system's use of any particular database or rule-engine library. What is explicitly not claimed as novel includes generic retrieval-augmented generation, large language model use for compliance reasoning in general, deterministic rule engines, rule extraction as a technique, a hybrid rule-plus-LLM architecture in general, and generic LLM-to-LLM routing.

TABLE IX
NOVELTY AND CONTRIBUTION ANALYSIS.

Claim	Verdict	Basis
Hybrid rule-engine + RAG + LLM architecture for compliance	Not novel in isolation	Established pattern in recent work: knowledge-graph-augmented RAG compliance [5], neuro-symbolic compliance reasoning [6], compliance-as-code [4], and code-generation-based compliance checking [8].
Automated extraction of rules from regulatory text	Not novel as a technique; novel as an evaluated RBI-specific instance	A well-established lineage in automated compliance checking for construction codes [1]–[3]; no equivalent RBI-evaluated system located (Section III.D).
Adaptive reasoning-modality router with a formal cost objective, evaluated against routing baselines	**Primary defensible contribution**	Structurally distinct from the populated LLM-model-routing literature [16]–[22], which routes between LLMs rather than between reasoning modalities; not found applied to compliance-modality selection.
A single regulatory knowledge base reused by both a human-facing regulation comprehension interface and a machine-facing compliance decision engine	System-level contribution supporting the primary claim	Not itself claimed as a new technique; supports deployability of the router (Section X).
Rule-grounded synthetic RBI compliance benchmark with graded case-difficulty categories, intended for public release	Defensible secondary contribution	No public RBI-specific benchmark located; construction methodology aligned with recent calls for richer benchmark case coverage [23]–[25].
Connected-counterparty graph reasoning for exposure aggregation	Not claimed as a validated contribution	Evaluated only on a small synthetic subset (8 test cases) resting on ownership data generated by the authors, which is insufficient to support a general result (Sections IV.A, XVI, XX).
RBI-specific regulation comprehension (structured extraction, versioning, citation-grounded QA over Master Directions/circulars)	Not novel as a technique; novel as an RBI-evaluated, jointly-integrated instance	Retrieval-augmented generation [9], layout-aware document understanding [10], and RAG evaluation methodology [11] are all established; no equivalent system evaluated jointly on RBI comprehension and RBI compliance checking located (Section III.D).

XIX. LIMITATIONS AND THREATS TO VALIDITY

A. Threats to validity

Rule-extraction verification is designed to evaluate on a bounded, disclosed sample (Section XII) and therefore extraction accuracy figures obtained on it should not be extrapolated to apply universally to all RBI regulation. Large language model choice impacts reproducibility: extraction and retrieval-augmented-reasoning results are anticipated to change if the underlying model is updated or replaced; model identity and version are pinned and reported (Section XII) precisely so that this dependency is auditable rather than hidden. The router's learned component is trained and tuned on the same benchmark family used for evaluation, using a held-out validation split; results should be interpreted as within-benchmark generalization, not as a generalization claim to domains outside RBI banking regulation.

B. Scope of the empirical evidence

The experimental evidence in Section XVI is limited to the compliance-checking module on a synthetic benchmark whose test partition contains 294 cases; several test subgroups are small (for example, 10 cross-version, 12 category-B, and 8 category-G cases), results come from a single rule-level split, and confidence intervals and significance tests are reported only for the main router comparisons. Results of the Regulation Comprehension benchmark, citation accuracy, hallucination rate, latency, and per-category rule-extraction accuracy are not reported, so RQ3, RQ4, RQ6, and RQ7 remain unanswered.

C. No real operational bank data

All quantitative results reported in this paper were obtained on the synthetic benchmark of Section XII, not on real bank operational data, since such data is confidential, restricted by regulations, and not available for research. Every result must be read as a controlled experiment of the reasoning behavior of the system against known, rule-based ground truth, and not as a statement on the accuracy of the system's output on a real-world bank's operational data. This limit is a property of the problem domain, not of this system, and is shared by essentially any academic work in this space that cannot obtain a data-sharing agreement with a regulated financial institution.

D. Bounded rule-extraction verification and scope exclusions

The connected-counterparty graph reasoning (category G of Section VI) is evaluated only on a small synthetic subset (Section XVI) and is not validated against real ownership data. Optical character recognition and scanned-document ingestion are outside the evaluated scope (Section IV.A). Both are discussed as future work in Section XX, rather than presented as implemented capabilities.

E. Regulatory drift and deployment considerations

RBI regulation is amended and superseded on an ongoing basis. The version, effective_from, and effective_to fields, along with the cross-version benchmark cases (Section XII), test whether the system's versioning logic behaves correctly on known amendments, but a deployed system would, in addition, require a live re-ingestion pipeline to detect and process new amendments as they are published—out of scope for this paper's evaluation, which operates on a fixed regulatory snapshot.

F. Security and privacy considerations

A system working with bank-submitted documents in a production setting would have to handle prompt-injection risk from maliciously or adversarially crafted submitted documents, and would need controls around data leakage of confidential bank information through logging, retrieval indices, or LLM API calls. These are deployment-time engineering concerns rather than properties evaluated in this research paper, and are noted here for completeness rather than addressed empirically.

XX. Future Work

Realistic extensions beyond what was considered in this paper could include running the comprehension evaluation protocol of Sections XIII and XV-B and reporting its results, and extending the compliance evaluation with confidence intervals and larger test partitions; acquiring real bank operational data through secure institutional partnerships while respecting appropriate confidentiality and regulatory approval, in order to validate results beyond the synthetic benchmark of Section XII; a live RBI regulatory-update pipeline to detect and re-ingest amendments as they are published, which would extend the offline processing of Section XI; better processing of scanned or image-based regulatory or evidence documents using optical character recognition, currently not included (Section IV.A); multilingual regulation comprehension, extending the analysis of Section V to non-English regulatory or evidentiary text; connected-counterparty graph reasoning to aggregate exposure (category G, Section VI), illustrated conceptually below; a richer regulation knowledge graph that captures cross-document relationships beyond the cross-reference handling already described in Section V; continuous learning for the router's cost-minimizing policy as new labeled routing outcomes accumulate in deployment; stronger human-in-the-loop workflows for the REQUIRES_HUMAN_REVIEW path, including reviewer feedback that updates historical_router_accuracy (Table V); secure architecture for confidential bank data to address the concerns noted in Section XIX.F; and broader coverage of Indian financial regulation beyond RBI banking regulation, such as SEBI or IRDAI requirements, where the underlying architecture might transfer but has not been evaluated.

A. Connected-counterparty graph reasoning (theoretical sketch)

RBI's large-exposure framework requires aggregating exposure across connected counterparties—parent/subsidiary, common ownership, guarantee, and economic-dependency relationships. This subsystem is evaluated in this work only on a small synthetic subset (Section XVI). As a theoretical sketch of a fuller treatment: entities and relationships would be represented as a property graph, with nodes as legal entities and edges as ownership, control, or guarantee relationships weighted by ownership percentage where available; connected-component or community-detection algorithms would identify counterparty groups; exposure-aggregation rules (categories F and G of Section VI) would then apply the existing quantitative rule logic to the aggregated group rather than the individual entity. A fuller evaluation would require multi-hop ownership data not publicly available, and evaluating the subsystem against ownership data generated by the authors themselves would not constitute independent validation. This is flagged explicitly as future work requiring either a data-sharing arrangement with a financial institution or a carefully justified, separately validated synthetic-graph methodology beyond the scope of this paper.

XXI. Conclusion

This paper presents an integrated regulatory intelligence system for RBI banking regulation, comprising a Regulation Comprehension module that transforms RBI circulars and Master Directions into searchable, citation-grounded regulatory knowledge, and an Adaptive Compliance Checking module that reuses this same knowledge to assess bank-submitted evidence from RBI's supervisory perspective. The system's value lies in a reasoning-modality router that chooses between a deterministic rule engine, a retrieval-based large language model, and human escalation, under an explicit cost–accuracy–latency objective, which we contrast with the well-established LLM-to-LLM model-routing literature. We support this with a schema-constrained rule-extraction pipeline for RBI regulatory text, a systematically constructed rule-grounded synthetic benchmark covering nine case-difficulty categories, and an evidence-based explanation architecture that tracks every decision and comprehension answer back to a particular rule, clause, and evidentiary or source input. We described the evaluation design of the system, comprising four compliance-checking baselines, three comprehension baselines, and ablations. On the held-out test partition of a synthetic benchmark, with LLM-path decisions produced by a surrogate judge, the adaptive router reached 92.2% automated and 96.9% end-to-end accuracy, above the fixed hybrid (85.4%), RAG+LLM (72.5%), and LLM-only (56.8%) baselines, and its operating points lay on the accuracy–cost Pareto frontier (Section XVI). Adversarial and ambiguous cases remained the weakest, the evidence is limited to synthetic data and a small test partition, and the comprehension-module results remain to be reported (Section XIX). We have been explicit throughout about what is and is not novel (Section XVIII) and about what is not validated, including connected-counterparty graph reasoning beyond a small synthetic subset (Sections XVI and XX) and optical character recognition (Section IV.A), on the view that a narrower, honestly evaluated contribution is more defensible than a broader, under-evaluated one.

Declarations

Funding
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Ethics, Consent to Participate, and Consent to Publish declarations: Not applicable.

References

[1] J. Zhang and N. M. El-Gohary, "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, p. 04015014, 2016.

[2] X. Xu and H. Cai, "Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure," *Advanced Engineering Informatics*, vol. 48, p. 101288, 2021.

[3] H. Hettiarachchi, A. Dridi, M. M. Gaber, *et al.*, "CODE-ACCORD: A corpus of building regulatory data for rule generation towards automatic compliance checking," *Scientific Data*, vol. 12, no. 1, p. 170, 2025.

[4] V. D. Ershov, "A case study for compliance as code with graphs and language models: Public release of the regulatory knowledge graph," arXiv:2302.01842, 2023.

[5] D. Guo, J. Wu, and S. M. Yiu, "ComplianceNLP: Knowledge-graph-augmented RAG for multi-framework regulatory gap detection," arXiv:2604.23585, 2026.

[6] Y.-S. Hsia, F. Yu, and J.-H. R. Jiang, "Neuro-symbolic compliance: Integrating LLMs and SMT solvers for automated financial legal analysis," arXiv:2601.06181, 2026.

[7] K. Khanvilkar and K. Kommuru, "Regulatory graphs and GenAI for real-time transaction monitoring and compliance explanation in banking," arXiv:2506.01093, 2025.

[8] S. Li, J. Chen, R. Yao, *et al.*, "Compliance-to-Code: Enhancing financial compliance checking via code generation," arXiv:2505.19804, 2025.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[10] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Pre-training of text and layout for document image understanding," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2020, pp. 1192–1200.

[11] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. European Chapter Assoc. Comput. Linguistics: System Demonstrations (EACL)*, 2024, pp. 150–158.

[12] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[13] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An expert-annotated NLP dataset for legal contract review," in *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[14] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Y. Wang, "FinQA: A dataset of numerical reasoning over financial data," in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 3697–3711.

[15] Commercial RBI compliance / GRC platform documentation (product literature, non-peer-reviewed), consulted for domain-application context only.

[16] D. Ding, A. Mallick, C. Wang, *et al.*, "Hybrid LLM: Cost-efficient and quality-aware query routing," in *Proc. ICLR*, 2024.

[17] D. Ding, A. Mallick, S. Zhang, *et al.*, "BEST-Route: Adaptive LLM routing with test-time optimal compute," in *Proc. ICML*, 2025.

[18] I. Ong, A. Almahairi, V. Wu, *et al.*, "RouteLLM: Learning to route LLMs from preference data," in *Proc. ICLR*, 2025.

[19] T. Feng, Y. Shen, and J. You, "GraphRouter: A graph-based router for LLM selections," in *Proc. ICLR*, 2025.

[20] W. Jitkrittum, H. Narasimhan, A. S. Rawat, *et al.*, "Universal model routing for efficient LLM inference," arXiv:2502.08773, 2025.

[21] L. Chen, M. Zaharia, and J. Zou, "FrugalGPT: How to use large language models while reducing cost and improving performance," *Transactions on Machine Learning Research*, 2024.

[22] Y. Moslem and J. D. Kelleher, "Dynamic model routing and cascading for efficient LLM inference: A survey," arXiv:2603.04445, 2026.

[23] B. Payzun *et al.*, "Benchmarks are not validation: A system-level view of financial LLM applications," arXiv:2607.28840, 2026.

[24] D. Song, Y. Huang, *et al.*, "Evaluating implicit regulatory compliance in LLM tool invocation via logic-guided synthesis," arXiv:2601.08196, 2026.

[25] G. Matlin, A. Theerthala, *et al.*, "FinForge: Semi-synthetic financial benchmark generation," arXiv:2601.06747, 2026.

[26] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904.

[27] N. Guha, *et al.*, "LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

[28] I. Chalkidis, *et al.*, "LexGLUE: A benchmark dataset for legal language understanding in English," in *Proc. 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022, pp. 4310–4330.

[29] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing statistical machine translation for text simplification," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2016.